# Guide to setup user environment in Union VMS on local computer

This is a guide to setup and install tools to run **Union** VMS on local computer.

# SoapUI

1. Download the open source version at <http://www.soapui.org/downloads/soapui.html>
2. Install the SoapUI-x64-x.x.x.exe file and make sure that HermesJMS is selected during the installation process.

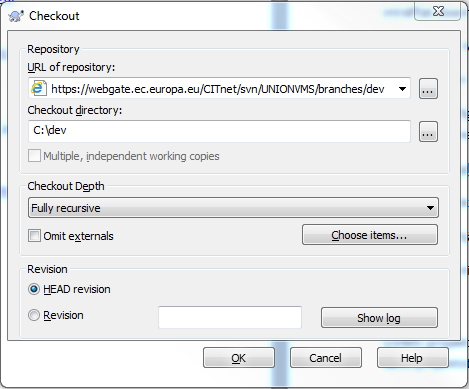
Installation Guide is found at <http://www.soapui.org/getting-started/installing-soapui/installing-on-windows.html>.

# Subversion (Tortoise SVN)

1. Download TortoiseSvn at <https://tortoisesvn.net/downloads.html>
2. Install the TortoiseSVN-1.9.3.27038-x64-svn-X.X.X.msi file.

More information about is found at <https://tortoisesvn.net/docs/nightly/TortoiseSVN_en/tsvn-quick-start.html#id493001>

# Get latest version of Union VMS software to local Computer

1. Create folder c:\dev on the computer.
2. Right-Click on the dev folder and select “SVN checkout”
3. Paste the link <https://webgate.ec.europa.eu/CITnet/svn/UNIONVMS/branches/dev> to the “URL of repository” according to following figure and press ENTER:  
     
     
     
   Note: We reference to the dev branch in this user guide.

# Install JDK (Java Development Kit)

1. Download the JDK8 here <http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>.
2. Install the jdk-8uXX-windows-x64.exe.
3. Make sure that JAVA\_HOME is set in your environment variables. To do this do following:
   1. Click on the startbutton.
   2. Right click on "Computer"
   3. Choose properties
   4. Choose Advanced settings
   5. Click on the button "Environment variables"
   6. Check that JAVA\_HOME is present in the System variables
      1. If not, press New
      2. Add JAVA\_HOME as the variable name
      3. Add the path to your JDK folder in the value input   
         (For example C:\Program Files\Java\jdk1.8.0)

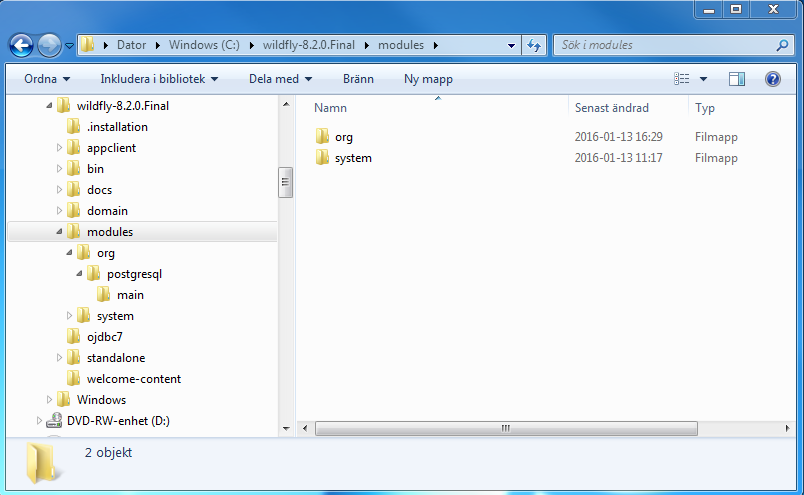
# Install Maven

1. Download Maven at <http://maven.apache.org/download.cgi>.
2. The info below is taken from <http://maven.apache.org/install.html>
   1. Unzip the distribution archive, i.e. *apache-maven-3.3.9-bin.zip* to the directory you wish to install Maven 3.3.9. These instructions assume you choose *C:\Program Files\Apache Software Foundation*. The subdirectory apache-maven-3.3.9 will be created from the archive.
   2. Add the unpacked distribution's bin directory to your user *PATH* environment variable by opening up the system properties (WinKey + Pause), selecting the "Advanced" tab, and the "Environment Variables" button, then adding or selecting the *PATH* variable in the user variables with the value C:\Program Files\Apache Software Foundation\apache-maven-3.3.9\bin.
   3. **Optional**: In the same dialog, add the MAVEN\_OPTS environment variable in the user variables to specify JVM properties, e.g. the value -Xms256m -Xmx512m. This environment variable can be used to supply extra options to Maven.
   4. In the same dialog, make sure that JAVA\_HOME exists in your user variables or in the system variables and it is set to the location of your JDK, e.g. C:\Program Files\Java\jdk1.8.0\_66.
   5. Open a *new* command prompt (Winkey + R then type cmd) and run mvn --version to verify that it is correctly installed.
   6. Download settings.xml at <https://projektplatsen.havochvatten.se/share/page/site/union-vms/documentlibrary#filter=path%7C%2FProject%2520Workspace%2FDevelopment%2FEnvironment%2520setup%7C&page=1>
   7. Replace exist file settings.xml found in C:\Program Files\Apache Software Foundation\apache-maven-3.3.9\conf with the download file.
   8. Modify settings.xml in path C:\Program Files\Apache Software Foundation\apache-maven-3.3.9\conf according to following:
      1. Change following line from   
           
          <localRepository>C:\Users\jojoha\.m2\repository</localRepository>  
           
         to  
           
          <localRepository>C:\Users\<YOUR USER-ID>\.m2\repository</localRepository>
      2. Change following lines from   
           
          <servers>  
          <server>   
          <id>hav.releases</id>   
          <username>deployment</username>   
          <password>deployment123</password>   
          </server>   
          <server>   
          <id>hav.snapshots</id>   
          <username>deployment</username>   
          <password>deployment123</password>   
          </server>   
          </servers>  
           
         to  
           
          <servers>  
          <server>   
          <id>hav.releases</id>   
          <username>deployment</username>   
          <password>deployment123</password>   
          </server>   
          <server>   
          <id>hav.snapshots</id>   
          <username>deployment</username>   
          <password>deployment123</password>   
          </server>   
          <server>   
          <id>digit-releases</id>   
          <username>emglemnmi</username>  
          <password>YOUR PASSWORD</password>  
          </server>   
          <server>   
          <id>digit-snapshots</id>   
          <username>emglemnmi</username>  
          <password>YOUR PASSWORD</password>  
          </server>   
          </servers>
      3. Change following lines from   
           
          <mirrors>  
          <mirror>  
          <id>hav-mirror</id>  
          <name>hav-central</name>  
          <url>http://svn:9080/nexus/content/groups/public/</url>  
          <mirrorOf>central</mirrorOf>  
          </mirror>  
          </mirrors>  
           
         to  
           
          <mirrors>  
          <mirror>  
          <id>hav-mirror</id>  
          <name>hav-central</name>  
          <url>http://svn:9080/nexus/content/groups/public/</url>  
          <mirrorOf>central</mirrorOf>  
          </mirror>  
          <mirror>  
          <id>mare-mirror</id>  
          <name>hav-central</name>  
          <url>http://livm06p:9080/nexus/content/groups/unionvms/</url>  
          <mirrorOf>central</mirrorOf>  
          </mirror>  
          </mirrors>
      4. Change following lines from   
           
          <mirrors>  
          <mirror>  
          <id>hav-mirror</id>  
          <name>hav-central</name>  
          <url>http://svn:9080/nexus/content/groups/public/</url>  
          <mirrorOf>central</mirrorOf>  
          </mirror>  
          </mirrors>  
           
         to  
           
          <mirrors>  
          <mirror>  
          <id>hav-mirror</id>  
          <name>hav-central</name>  
          <url>http://svn:9080/nexus/content/groups/public/</url>  
          <mirrorOf>central</mirrorOf>  
          </mirror>  
          <mirror>  
          <id>mare-mirror</id>  
          <name>hav-central</name>  
          <url>http://livm06p:9080/nexus/content/groups/unionvms/</url>  
          <mirrorOf>central</mirrorOf>  
          </mirror>  
          </mirrors>
      5. Change following lines from   
           
          <profiles>  
            
          <profile>  
          <id>default</id>  
          <activation>  
          <activeByDefault>true</activeByDefault>  
          </activation>  
          <repositories>  
          <repository>  
          <id>hav.releases</id>  
          <name>HaV release repo</name>  
          <url>[http://svn:9080/nexus/content/repositories/releases/</url](http://svn:9080/nexus/content/repositories/releases/%3c/url)>  
          <releases>  
          <enabled>true</enabled>  
          </releases>  
          <snapshots>  
          <enabled>false</enabled>  
          </snapshots>  
          </repository>  
          <repository>  
          <id>hav.snapshots</id>  
          <name>HaV snapshot repo</name>  
          <url>http://svn:9080/nexus/content/repositories/snapshots/</url>  
          <releases>  
          <enabled>false</enabled>  
          </releases>  
          <snapshots>  
          <enabled>true</enabled>  
          </snapshots>  
          </repository>  
          </repositories>  
          </profile>  
            
          </profiles>  
           
           
         to  
           
          <profiles>  
            
          <profile>  
          <id>default</id>  
          <activation>  
          <activeByDefault>true</activeByDefault>  
          </activation>  
          <repositories>  
          <repository>  
          <id>hav.releases</id>  
          <name>HaV release repo</name>  
          <url>[http://svn:9080/nexus/content/repositories/releases/</url](http://svn:9080/nexus/content/repositories/releases/%3c/url)>  
          <releases>  
          <enabled>true</enabled>  
          </releases>  
          <snapshots>  
          <enabled>false</enabled>  
          </snapshots>  
          </repository>  
          <repository>  
          <id>hav.snapshots</id>  
          <name>HaV snapshot repo</name>  
          <url><http://svn:9080/nexus/content/repositories/snapshots/></url>  
          <releases>  
          <enabled>false</enabled>  
          </releases>  
          <snapshots>  
          <enabled>true</enabled>  
          </snapshots>  
          </repository>  
          </repositories>  
          </profile>  
          <profile>  
          <id>mare</id>  
          <activation>  
          <activeByDefault>true</activeByDefault>  
          </activation>  
          <repositories>  
          <repository>  
          <id>digit-releases</id>  
          <name>DG DIGIT Repository</name>  
          <url>https://webgate.ec.europa.eu/CITnet/nexus/content/repositories/mare/</url>  
          <releases>  
          <enabled>true</enabled>  
          </releases>  
          <snapshots>  
          <enabled>false</enabled>  
          </snapshots>  
          </repository>  
          <repository>  
          <id>digit-snapshots</id>  
          <name>DG DIGIT Snapshot Repository</name>  
          <url>https://webgate.ec.europa.eu/CITnet/nexus/content/repositories/mare-snapshots/</url>  
          <releases>  
          <enabled>false</enabled>  
          </releases>  
          <snapshots>  
          <enabled>true</enabled>  
          </snapshots>  
          </repository>  
          </repositories>  
          </profile>  
            
          </profiles>

# ActiveMQ

1. Download ActiveMQ, for instance 5.11.1 at <http://activemq.apache.org/activemq-5111-release.html>.
2. Extract the zip to a path of your choice.  
   **NOTE: The path to the extracted files may NOT include space characters. The ActiveMQ software may not start then.**
3. Start the ActiveMQ server by doing following:
   1. Start a cmd window.
   2. cd to <Your path to activeMQ>\bin.
   3. Run the command   
      activemq start.
4. Monitor the queues in the ActiveMQ Console by go to link <http://localhost:8161/admin/index.jsp>.  
   NOTE: To monitor the queues in dev environment use following link:  
   <http://livm73u.hav.havochvatten.se:8161/admin/index.jsp>

# Install and setup Wildfly Application Server

1. Download Wilfly8.2.0-Final (Java EE7 Full & Web Distribution) at <http://wildfly.org/downloads/>.
2. Unzip the server to an appropriate folder where the wildfly server has read and write access.
3. Copy standalone-full.xml from the svn repo found under C:\dev\wildfly-config\ to <Your path>\wildfly-8.2.0.Final\standalone\configuration.
4. Modify standalone-full.xml according to following:
   1. Change all lines from   
        
       <driver>postgresql-9.3-1102.jdbc4.jar</driver>  
        
      to  
        
       <driver>postgresql</driver>
   2. Change all lines from  
        
       <drivers>  
       <driver name="h2" module="com.h2database.h2">  
       <xa-datasource-class>org.h2.jdbcx.JdbcDataSource</xa-datasource-class>  
       </driver>  
        
      to  
        
       <drivers>  
       <driver name="h2" module="com.h2database.h2">  
       <xa-datasource-class>org.h2.jdbcx.JdbcDataSource</xa-datasource-class>  
       </driver>  
       <driver name="postgresql" module="org.postgresql">  
       <xa-datasource-class>org.postgresql.xa.PGXADataSource</xa-datasource-class>  
       </driver>  
       </drivers>
   3. Change all lines from:  
        
       <connection-url>jdbc:postgresql://livmdb71u.havochvatten.se:5432/db71u</connection-url>  
        
      to  
        
       <connection-url>jdbc:postgresql://localhost:5432/db71u</connection-url>
5. Start cmd window and cd to <Your path>\wildfly-8.2.0.Final\bin
6. Run add-user.bat and select user type “a) Management User” and after that follow the instructions.
7. Locate C:\dev\wildfly-config\PostGis\wildfly\_spatial\_setup.zip
8. Extract the zip file to a folder of your choice.
9. Copy the files hibernate-spatial-4.3.jar and jts-1.13.jar to <Your path>\wildfly-8.2.0.Final\modules\system\layers\base\org\hibernate\main.
10. Modify module.xml in path <Your path>\wildfly-8.2.0.Final\modules\system\layers\base\org\hibernate\main by adding following lines:  
      
    <resource-root path="hibernate-spatial-4.3.jar"/>  
    <resource-root path="jts-1.13.jar"/>  
    <module name="org.postgresql"/>  
      
    In other words do like following example:  
    Change following lines from :  
      
     <resources>  
     <resource-root path="hibernate-core-4.3.7.Final.jar"/>  
     <resource-root path="hibernate-envers-4.3.7.Final.jar"/>  
     <resource-root path="hibernate-entitymanager-4.3.7.Final.jar"/>  
     <resource-root path="hibernate-infinispan-4.3.7.Final.jar"/>  
     <resource-root path="jipijapa-hibernate4-3-1.0.1.Final.jar"/>  
     </resources>  
      
     <dependencies>  
     <module name="asm.asm"/>  
     <module name="com.fasterxml.classmate"/>  
     <module name="javax.api"/>  
     <module name="javax.annotation.api"/>  
     <module name="javax.enterprise.api"/>  
     <module name="javax.persistence.api"/>  
     <module name="javax.transaction.api"/>  
     <module name="javax.validation.api"/>  
     <module name="javax.xml.bind.api"/>  
     <module name="org.antlr"/>  
     <module name="org.apache.commons.collections"/>  
     <module name="org.dom4j"/>  
     <module name="org.infinispan" services="import"/>  
     <module name="org.javassist"/>  
     <module name="org.jboss.as.clustering.common"/>  
     <module name="org.jboss.as.clustering.jgroups"/>  
     <module name="org.jboss.as.clustering.infinispan"/>  
     <module name="org.jboss.as.jpa.spi"/>  
     <module name="org.jboss.jandex"/>  
     <module name="org.jboss.logging"/>  
     <module name="org.jboss.vfs"/>  
     <module name="org.hibernate.commons-annotations"/>  
     </dependencies>  
      
    to  
      
     <resources>  
     <resource-root path="hibernate-core-4.3.7.Final.jar"/>  
     <resource-root path="hibernate-envers-4.3.7.Final.jar"/>  
     <resource-root path="hibernate-entitymanager-4.3.7.Final.jar"/>  
     <resource-root path="hibernate-infinispan-4.3.7.Final.jar"/>  
     <resource-root path="jipijapa-hibernate4-3-1.0.1.Final.jar"/>  
     ***<resource-root path="hibernate-spatial-4.3.jar"/>  
     <resource-root path="jts-1.13.jar"/>*** </resources>  
      
     <dependencies>  
     <module name="asm.asm"/>  
     <module name="com.fasterxml.classmate"/>  
     <module name="javax.api"/>  
     <module name="javax.annotation.api"/>  
     <module name="javax.enterprise.api"/>  
     <module name="javax.persistence.api"/>  
     <module name="javax.transaction.api"/>  
     <module name="javax.validation.api"/>  
     <module name="javax.xml.bind.api"/>  
     <module name="org.antlr"/>  
     <module name="org.apache.commons.collections"/>  
     <module name="org.dom4j"/>  
     <module name="org.infinispan" services="import"/>  
     <module name="org.javassist"/>  
     <module name="org.jboss.as.clustering.common"/>  
     <module name="org.jboss.as.clustering.jgroups"/>  
     <module name="org.jboss.as.clustering.infinispan"/>  
     <module name="org.jboss.as.jpa.spi"/>  
     <module name="org.jboss.jandex"/>  
     <module name="org.jboss.logging"/>  
     <module name="org.jboss.vfs"/>  
     <module name="org.hibernate.commons-annotations"/>  
     ***<module name="org.postgresql"/>***  
     </dependencies>
11. Copy the folder **org** and ALL subfolders and files under that folder from the extracted wildfly\_spatial\_setup.zip to <Your path>\wildfly-8.2.0.Final\modules like following printout:  
      
    
12. Copy the file activemq-rar.rar located in C:\dev\wildfly-config folder to C:\wildfly-8.2.0.Final\standalone\deployments.
13. Start the ActiveMQ server by doing following:
    1. Start a cmd window.
    2. cd to <Your path to activeMQ>\bin.
    3. Run the command   
       activemq start.
14. Start to the wildfly application server with standalone-full-config, run following in a cmd window:  
      
    <Your path>\wildfly-8.2.0.Final\bin\standalone.bat --server-config=standalone-full.xml

# Setup for Oracle Database

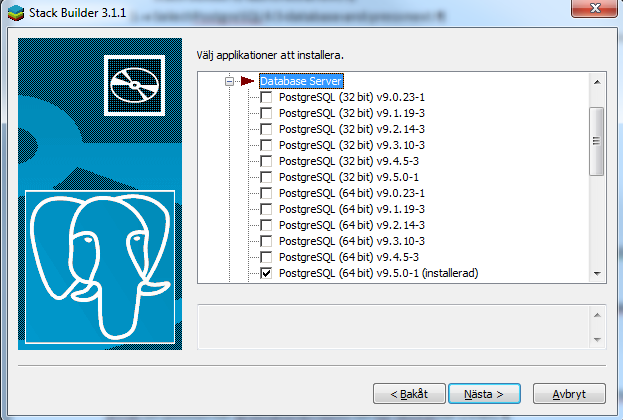
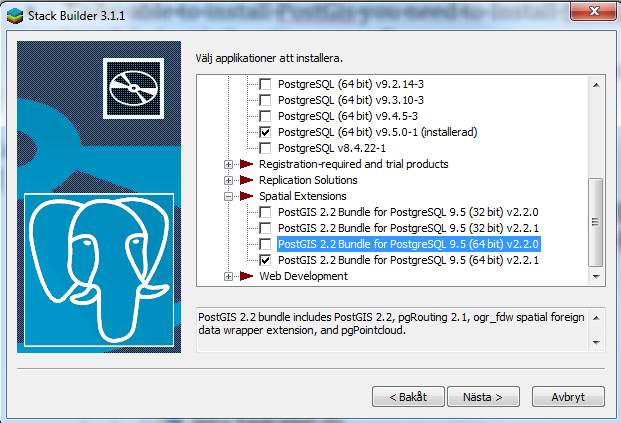
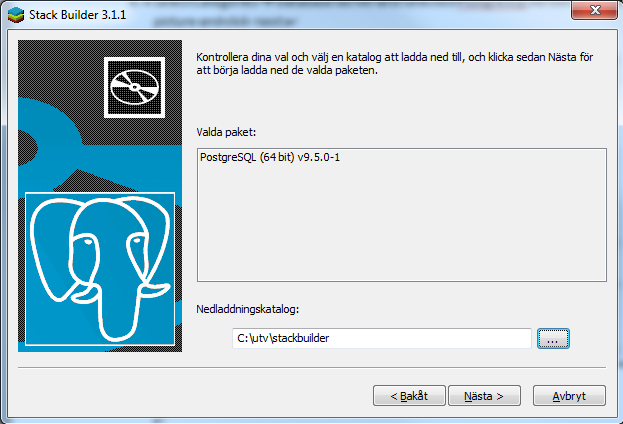
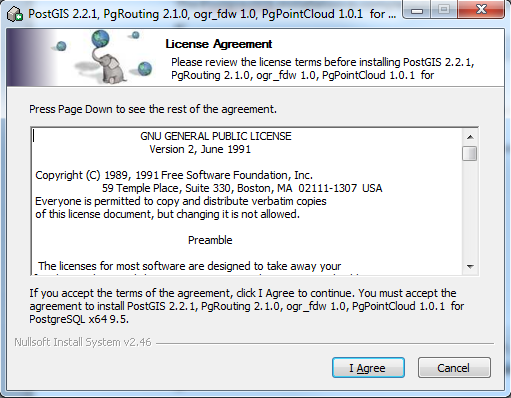
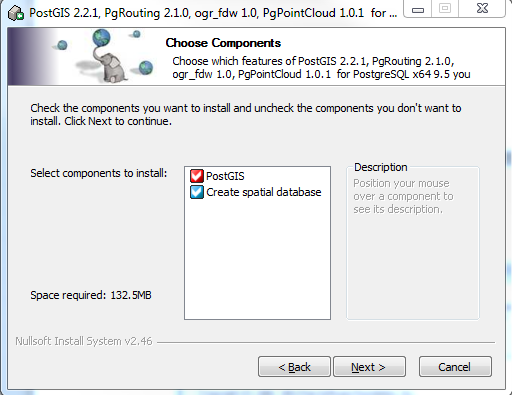
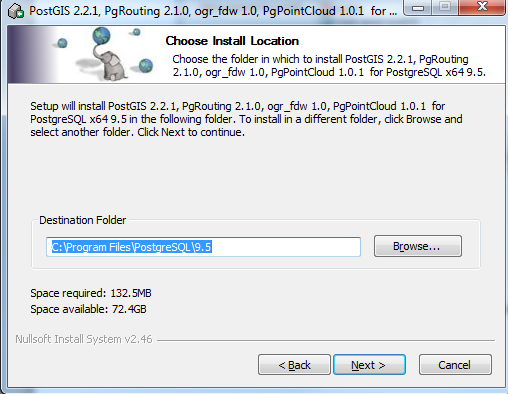
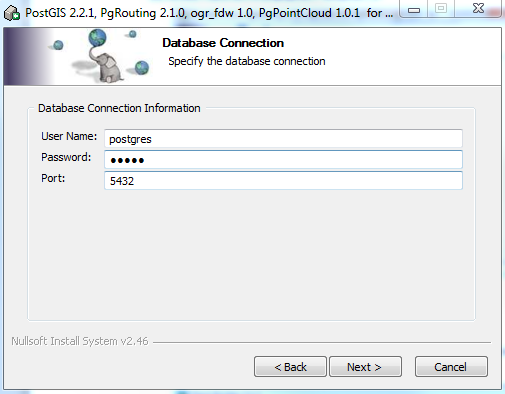
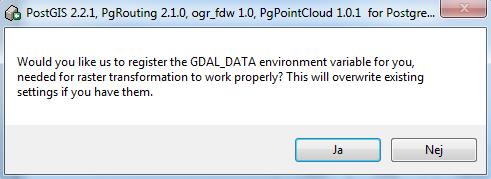
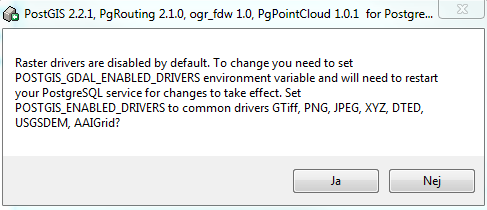
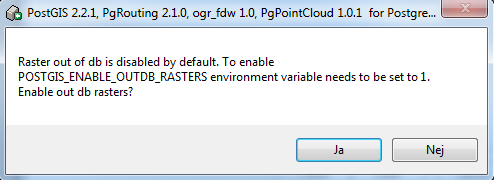
1. Copy the ojdbc7.jar from C:\dev\wildfly-config to a new folder of your choice.
2. Start up your Wildfly application server if not already started.
3. Start browser and go to link <http://localhost:9990/console/>.
4. Log in as user created in chapter” Install and setup Wildfly Application Server “step 6.
5. Choose Deployments.
6. Choose add and point to your ojdbc7.jar file on your local disk.
7. Choose next.
8. Click the "Enable" checkbox so the jar will enable when it is deployed.
9. Click Save.
10. Now your drivers for Oracle are installed and will deploy the next time your server starts.

# Add a Oracle datasource (Optional)

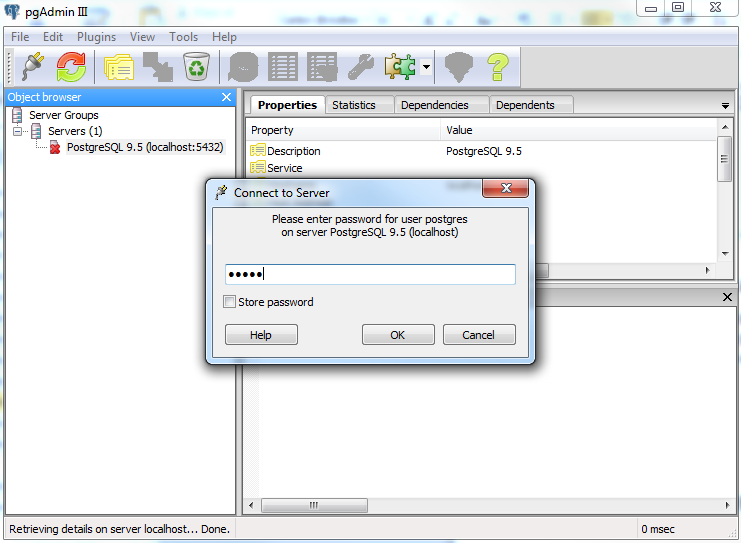
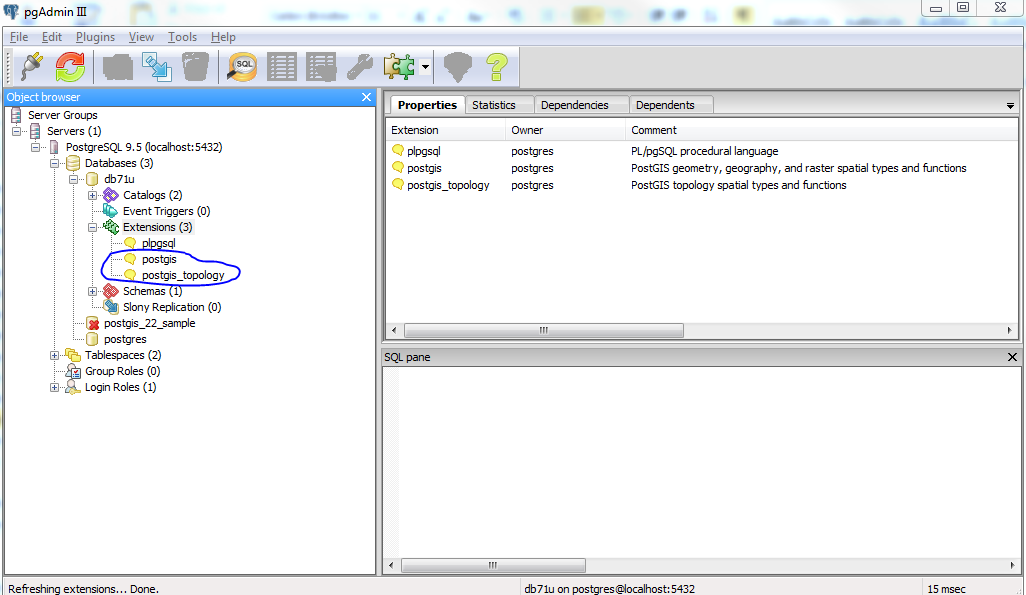
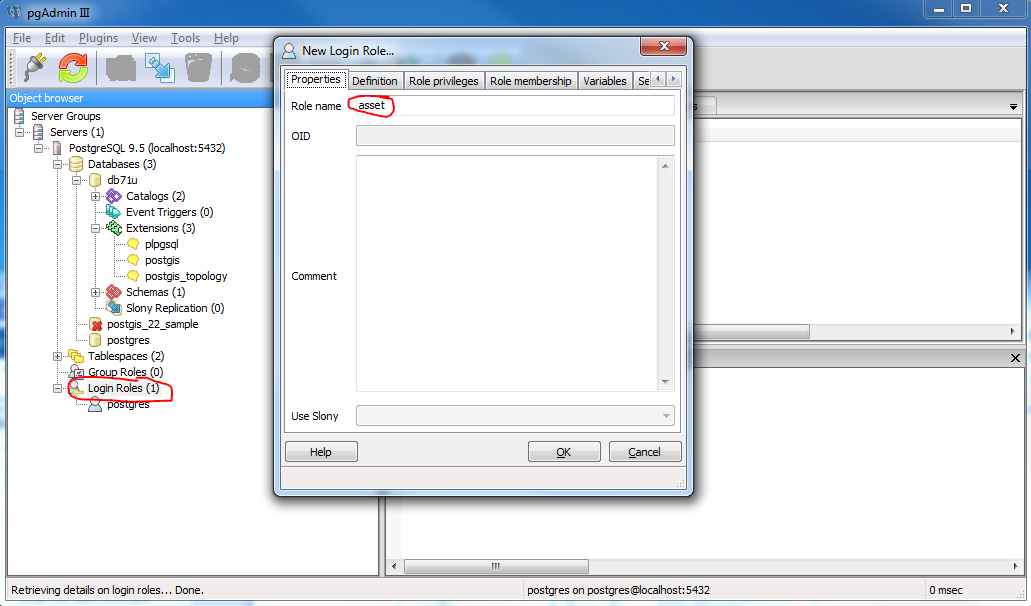
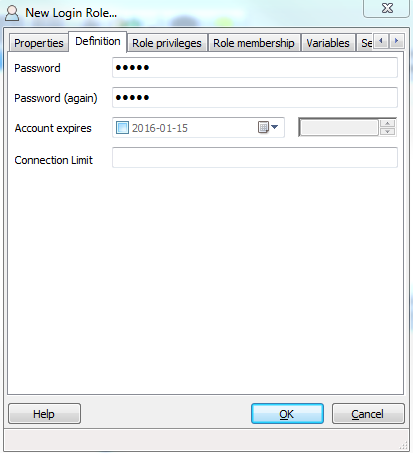
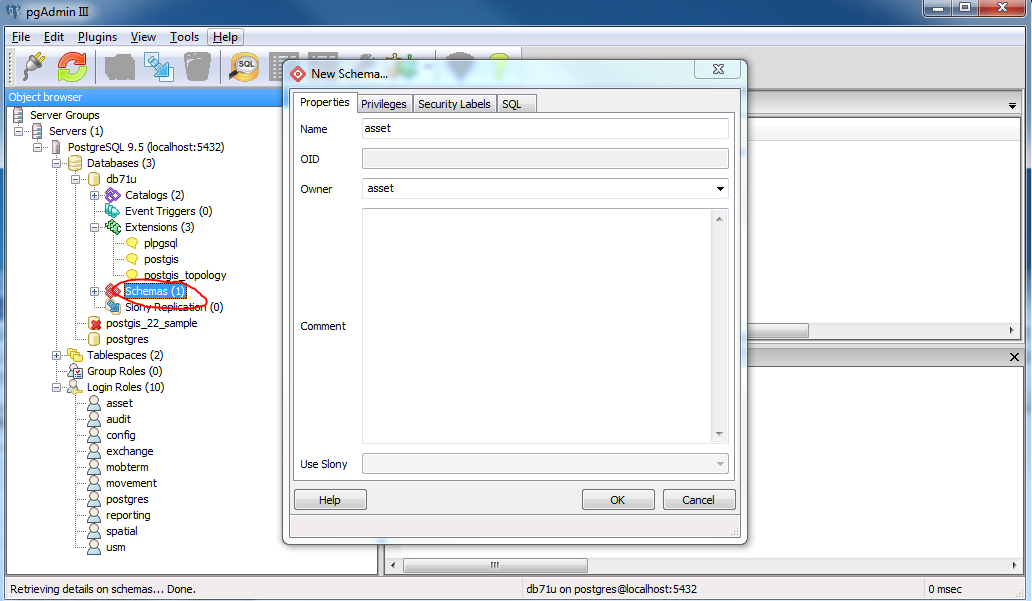
The modules in UnionVms will have self-configured default datasources for each module but if you want to add an own datasource you can do it like this. Another option is to add the datasource manually in the standalone-full.xml file.

1. Click Configuration.
2. Click on datasources
3. Datasource attributes
   1. The name is global and can be set to whatever you want
   2. The JNDI name should have the following namespace java:jboss/datasources/YourDataSource. This is what you put in your persistance.xml file in your java project for JTA
4. JDBC driver. If your driver is correctly installed according to the "Install the ojdbc driver" in the previous section the oracle driver should show up here. Choose it and click next.
5. Connection Settings. Put your connection settings here and test the connection. If you get a "Connection success" click Done and you have added your brand new datasource!

# Install PostgreSQL

1. Download the latest version of PostgreSQL at http://www.postgresql.org/download/windows/.
2. Install the postgresql-9.x.x-x-windows-x64.exe and follow the instructions. Make sure that Stack builder is launched at exit.
3. Select PostgreSQL 9.5 database and press next.
4. Select Categories 🡪 Database Server and selected PostgreSQL version AND under Spatial Extensions 🡪 PostGIS Bundle for PostgresSQL according to following pictures and click next:  
     
     
     
   
5. Select Download path for the packages and click next and follow the remaining installation instructions.  
     
   
6. When PostgresSQL setup is shown click next several times so that the PostgreSQL installation phase starts.  
     
     
     
   The PostgresSQL installation is completed when following printout is presented.   
     
   
7. Click on the finish button to close the PostgreSQL setup and continue with the stack builder installation.
8. The PostGIS installation setup starts.  
     
   
9. Select “Create spatial database and click next..  
     
   
10. Click next on installation location.  
    
11. Enter admin password created in earlier step in this chapter.  
    
12. Click on install.
13. Click on Yes if following pop up windows occurs.  
      
    
14. Click on Yes if following pop up windows occurs.  
      
    
15. Click on Yes if following pop up windows occurs.  
      
    
16. Click on close to finish PostGIS installation.
17. Click on the abort button in the stack builder setup and reboot the computer.  
      
    

# Setup and configure PostgresSQL database

1. After installation of PostgresSQL in the chapter before, Start the pgAdmin III application.
2. Double click on the PostSQL 9.5 (localhost:5432) according to following figure and enter the admin password created during installation of PostgresSQL in the chapter before.  
     
   
3. Right click on Databases and create a new database with name db71u and postgres as Owner and click OK.
4. Expand the db71u database and right click on Extensions.
5. Choose New Extension and in the dropdown name list choose Postgis.
6. Choose New Extension and in the dropdown name list choose Postgis\_topology. See picture below:  
     
   
7. Add user “asset” by right click on “Login Roles”. Enter role name “asset”.  
     
   
8. Click on the Definition tab and enter password “asset” (without quotes) in the password fields and click OK.  
     
   
9. Repeat step 8 for users:  
     
   audit  
   config  
   exchange  
   mobterm  
   movement  
   reporting  
   rules  
   spatial (Use “postgres” as password)  
   usm
10. Add scheme “asset” by right click on “Schemes”. Enter scheme name “asset” and select “asset” as owner.  
      
    
11. Repeat step 10 for users:  
      
    audit  
    config   
    exchange  
    mobileterminal  
    movement   
    reporting  
    rules  
    spatial  
    usm
12. To generate tables and test data for the asset module, start cmd window and cd to C:\dev\Modules\Asset\LIQUIBASE.
13. Run the command:  
      
    mvn liquibase:update -P postgres-local
14. Repeat step 12 and 13 for all remaining modules:  
      
    audit  
    config  
    exchange  
    mobileterminal  
    movement  
    reporting  
    rules  
    spatial  
    usm  
      
    **Note 1:** Change following line in the pom.xml from:  
      
    jdbc:postgresql://localhost:5432/uvms  
      
    to  
      
    jdbc:postgresql://localhost:5432/db71u  
      
    for modules Reporting, Spatial and   
      
      
    **Note 2: usm module is found under C:\trunk\USM\database\liquibase where C:\trunk is connected to** <https://webgate.ec.europa.eu/CITnet/svn/UNIONVMS/trunk>   
    The pom-xml under this path must be replaced with following pom.xml found under C:\dev\General documentation\Development BEFORE execution of step 13. (repository does not exist)  
      
    **Note 3:** Change the date to correct standard in file <Branches>\Modules\Spatial\LIQUIBASE\test\csv\user\_areas.csv For spatial module before executing step 13.

# Deploy modules to local Wildfly environment

1. Make sure that ActiveMQ and Wildfly is started according to step 13 and 14 in chapter “Install and setup Wildfly Application Server”.
2. Cd to Asset module by cd to C:\dev\Modules\Asset\APP.
3. Generate the wsdl classes for the Asset module by running following command:  
     
   mvn clean install -Pgenerate-from-wsdl
4. Compile the Asset module by running following command:  
     
   mvn clean install
5. Generate the REST API for the Asset module by running following command:  
     
   mvn clean install -P generate-rest-doc -Dhostname=localhost -Dport=8080
6. Cd to C:\dev\Modules\Asset\APP\module
7. Deploy asset module by excuting followning command:  
     
   mvn -P wildfly-deploy wildfly:deploy -Dhostname=127.0.0.1 -Dport=9990 -Dusername=admin -Dpassword=admin
8. Cd to Asset module by cd to C:\dev\Modules\Asset\DB.
9. Compile the Asset DB module by running following command:  
     
   mvn clean install
10. Cd to C:\dev\Modules\Asset\DB\module.
11. Deploy asset DB module by excuting followning command:  
      
    mvn -P wildfly-deploy wildfly:deploy -Dhostname=127.0.0.1 -Dport=9990 -Dusername=admin -Dpassword=admin
12. Repeat steps 2-11 for all remaining modules:  
      
    audit  
    config  
    exchange  
    mobileterminal  
    movement  
    reporting  
    rules  
    spatial  
    user  
      
    **Note:** Some of the module does not fully support standard. For instance reporting, spatial and User. These modules has no corresponding DB module which means that steps 8-11 shall not be executed for reporting and spatial.
13. To install Exchange plugin **sirusone** do following:  
    1. Cd to C:\dev\Modules\Exchange\PLUGIN\sirusone
    2. Compile the plugin module sirusone by running following command:  
         
       mvn clean install
    3. Cd to C:\dev\Modules\Exchange\PLUGIN\sirusone\module.
    4. Deploy the plugin module by excuting followning command:  
         
       mvn -P wildfly-deploy wildfly:deploy -Dhostname=127.0.0.1 -Dport=9990 -Dusername=admin -Dpassword=admin
14. Repeat step 13 for the plugin **twostage**.

# Application front end installation

1. Download “Node JS” at <https://nodejs.org/>
2. Install the downloaded node-vxxx.msi file.
3. Download “Git” at <https://git-scm.com/download/win>
4. Install the downloaded git-xxxx.exe file.   
   **Note:** Important to select **“Use Git from the Windows Command Prompt”** when prompted during the installation.
5. Start cmd window as administrator.
6. Run following command in the cmd window so npm dependencies gets installed:  
     
   cd C:\dev\unionvms-web  
   npm install
7. Run following commands in the cmd window:  
     
   cd C:\dev\unionvms-web  
     
   npm install –g grunt-cli bower karma  
     
   npm install -g phantomjs  
     
   npm install -g karma phantomjs-prebuilt karma-phantomjs-launcher

1. Run following command in the cmd window so bower dependencies gets installed:  
     
   cd C:\dev\unionvms-web  
   bower install
2. Modify Gruntfile.js according to following:  
     
   from  
     
    host: 'livm73u',  
    port: 28080  
     
   to  
     
    host: 'localhost',  
    port: 8080  
     
   AND  
     
   from  
     
    grunt.registerTask('build-local', ['test', 'clean:before', 'copy:configLocal', 'test', 'sub-build']);  
     
   to  
     
    grunt.registerTask('build-local', ['clean:before', 'copy:configLocal', 'sub-build']);
3. Check that folder font-awesome exists under C:\dev\unionvms-web\app\assets. If the folder does not exist copy font-awesome (including all sub folders and files) from C:\dev\unionvms-web\bower\_components to C:\dev\unionvms-web\app\assets.
4. Run following command in the cmd window so Grunt makes a build locally:  
     
   cd C:\dev\unionvms-web  
   grunt build-local
5. Start the front end server by running following command:  
     
   cd C:\dev\unionvms-web  
   grunt serve

# Web interface

## Wildfly

1. To access Wildfly through web interface go following URL:  
     
   <http://localhost:9990/console>
2. Enter user:admin with password:admin

## Grunt (Front end)

1. To access Grunt front-end through web interface go following URL:  
     
   http://localhost:9001/app
2. Enter user: vms\_admin\_comwith password:password