MOBILE APPLICATION TO SECURE TENURE

MAST Development Environment Installation Setup Guide

Abstract

This document provides detail of development environment setup, including the download and installation of various software/tools and source code.

Contents

USAID FROM THE AMERICAN PEOPLE

1 INTRODUCTION

This guide describes the setup of development environment of MAST web infrastructure and MAST android mobile application on windows environment. The objective of this document is to provide the detail of development environment setup, including the download and installation of various software and tools for end users so as to facilitate the recreation of MAST environment using the source code.

2 INSTALL AND CONFIGURE MAST DEVELOPMENT TOOLS

2.1 MAST Web Infrastructure

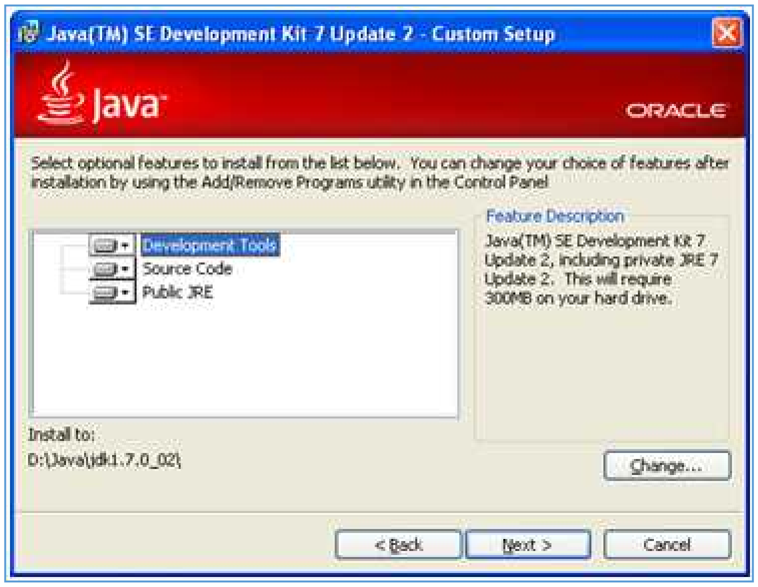
The core tools and software used to develop MAST web infrastructure are:

- Java JDK7
- PostgreSQL 9.3 /PostGIS 2.1
- Apache Tomcat 7
- Eclipse Luna IDE

You also need to download source code for MAST web infrastructure from GITHub (*https://github.com/MASTUSAID/MAST-DMI*) and minimal postgreSQL scripts (*https://github.com/MASTUSAID/DB-SCRIPTS*) that are to be run on the database to schema and load mandatory master tables.

2.1.1 Installing Java JDK 7

- Download the Java 7 JDK update 56 or later from http://www.oracle.com/technetwork/java/javase/downloads/index.html
- Run the exe file and follow the instructions.

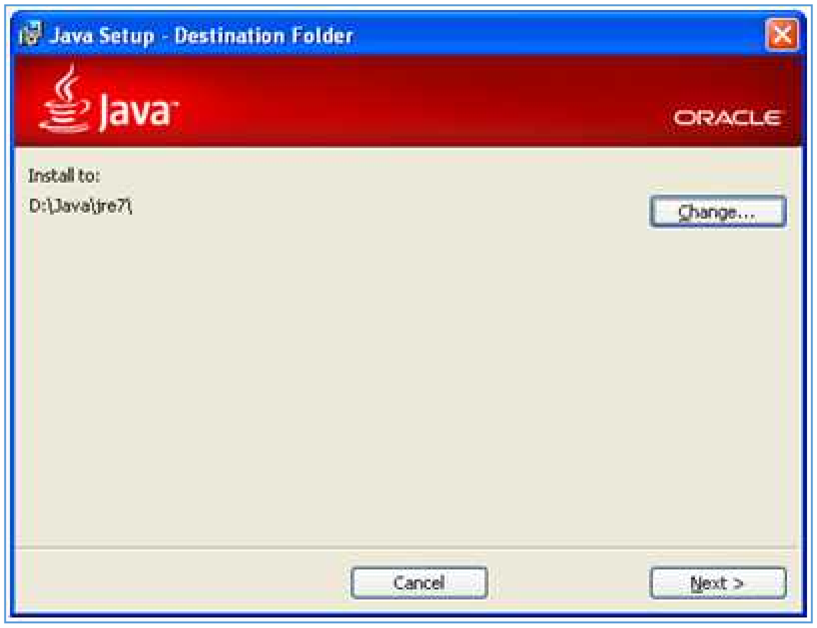


Select optional features to install from the list below. You can change your choice of features after installation by using the Add/Remove Programs utility in the Control Panel

Java(TM) SE Development Kit 7 Update 2, including private JRE 7 Update 2. This will require 300MB on your hard drive.

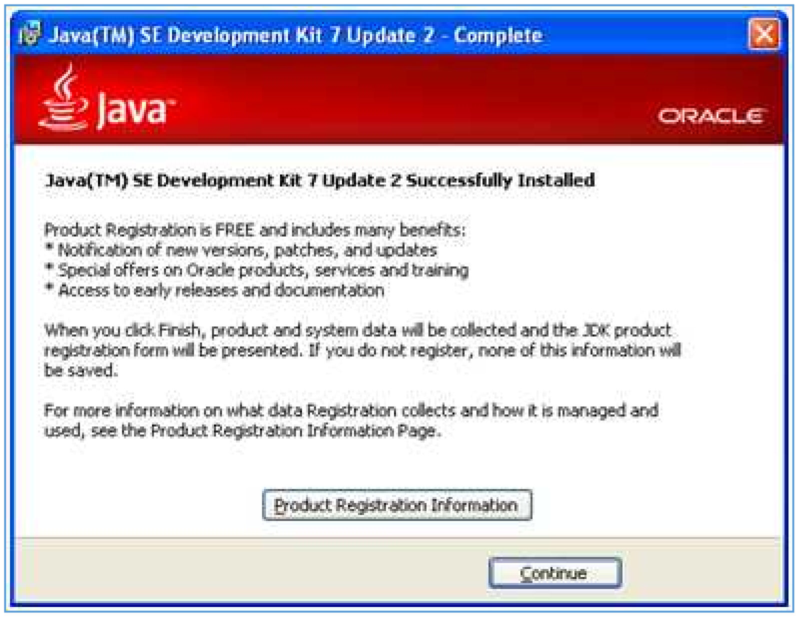
Install to:
D:\Java\jdk1.7.0_02\

- Change the install path as required and click next.
- Click Next >



Install to:
D:\Java\jre7\

- Installation complete.



Java(TM) SE Development Kit 7 Update 2 - Complete
Java
ORACLE
Product Registration Information
Continue

Java(TM) SE Development Kit 7 Update 2 Successfully Installed

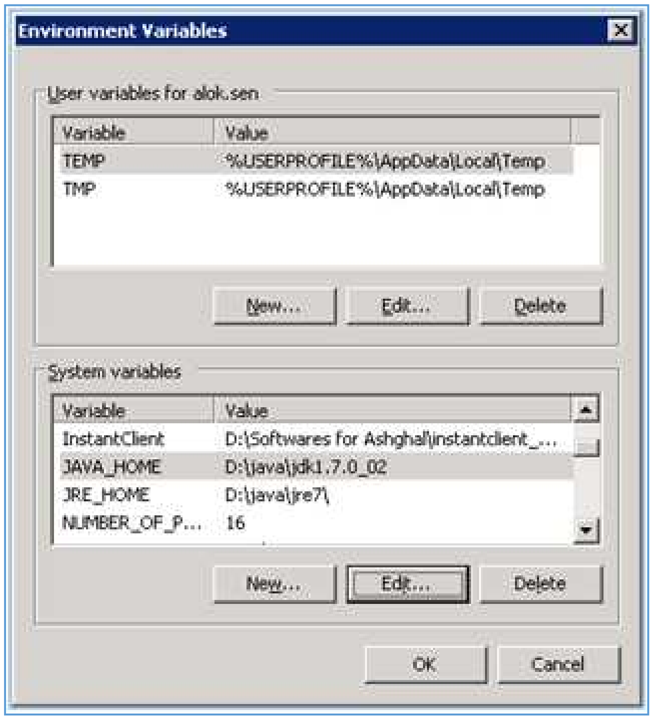
Product Registration is FREE and includes many benefits:
* Notification of new versions, patches, and updates
* Special offers on Oracle products, services and training
* Access to early releases and documentation

When you click Finish, product and system data will be collected and the JDK product registration form will be presented. If you do not register, none of this information will be saved.

For more information on what data Registration collects and how it is managed and used, see the Product Registration Information Page.

2.1.2 Setting Java and JRE Home

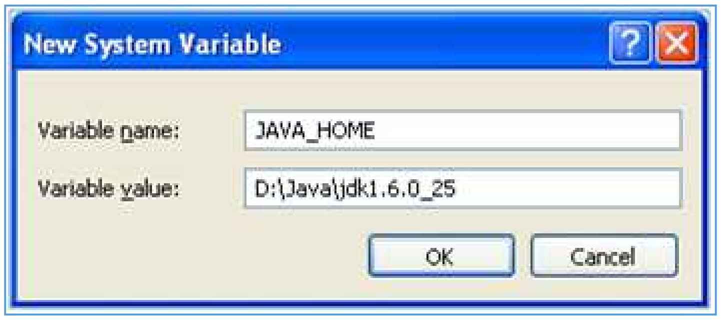
- From Control Panel open System Properties.
- Control Panel-> System Properties
- Click on Advanced Tab -> Environment Variables button.



Variable	Value
TEMP	%USERPROFILE%\AppData\Local\Temp
TMP	%USERPROFILE%\AppData\Local\Temp

Variable	Value
InstantClient	D:\Softwares for Ashghal\instantclient_...
JAVA_HOME	D:\java\jdk1.7.0_02
JRE_HOME	D:\java\jre7\
NUMBER_OF_P...	16

- In System variables click on new button.



Variable name: JAVA_HOME

Variable value: D:\Java\jdk1.6.0_25

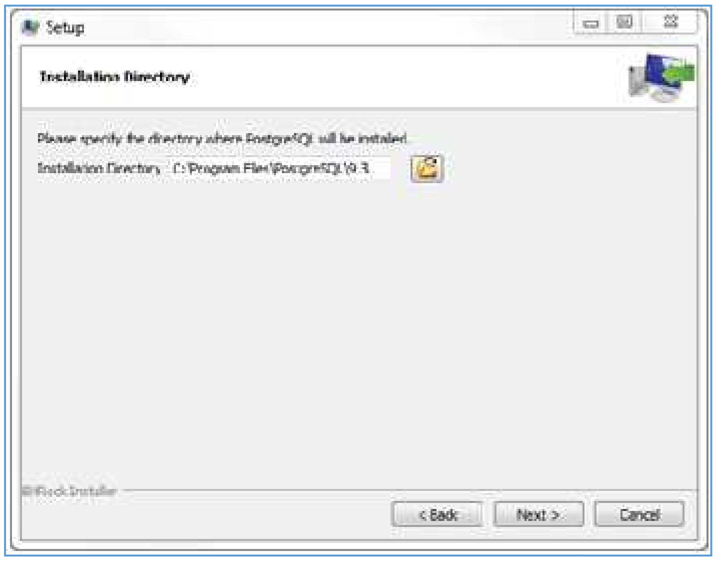
- Create a new system variable JAVA_HOME specifying the JDK installation folder as variable value.

- In the same way create JRE_HOME variable with JRE installation folder (ex. D:\java\jre7\).

2.1.3 Installing PostgreSQL

The graphical installer for PostgreSQL includes the PostgreSQL server, pgAdmin III, a graphical tool for managing and developing databases, and StackBuilder, a package manager that is used to download and install PostGIS.

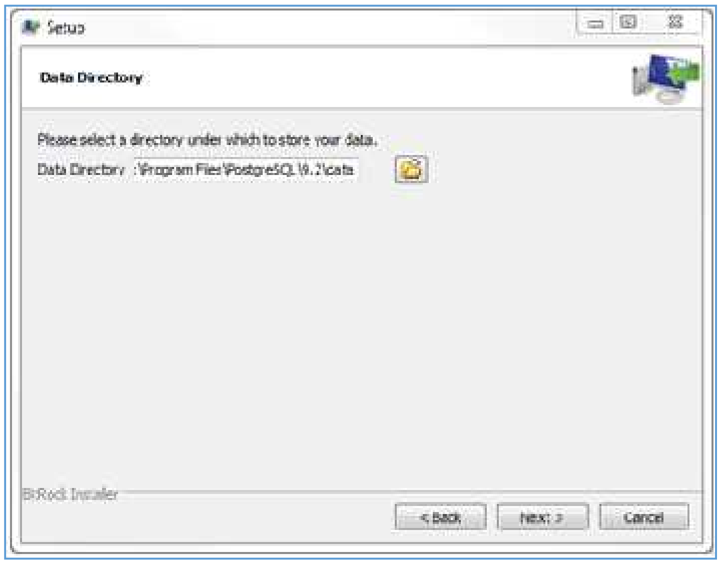
- Download postgreSQL graphic installer from http://www.enterprisedb.com/products-services-training/pgdownload#windows
- Double click on the installer file.
- Specify installation directory.


Installation Directory

Please specify the directory where PostgreSQL will be installed.

Installation Directory: C:\Program Files\PostgreSQL\9.3

- Specify the directory where you want to store the data in the database server.

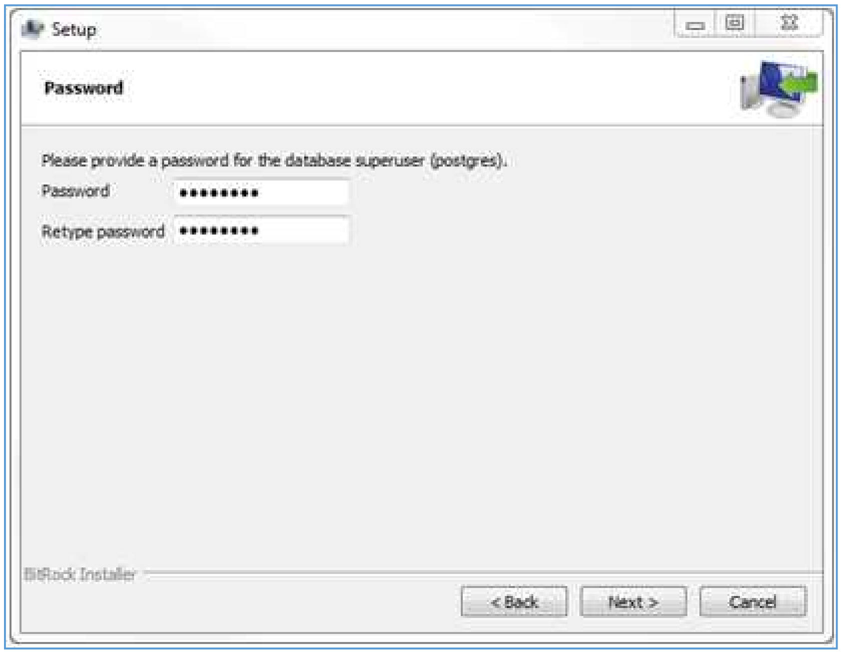

Setup

Data Directory

Please select a directory under which to store your data.

Data Directory : \Program Files\PostgreSQL\9.2\data

- Enter the password for the database superuser and service account. This password is set for default postgres user.


Setup

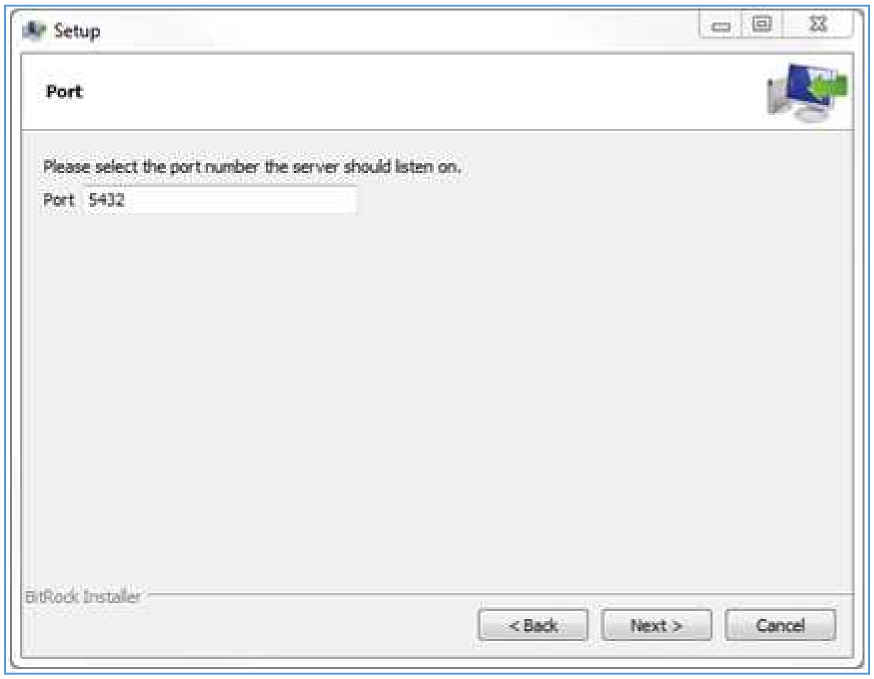
Password

Please provide a password for the database superuser (postgres).

Password ●●●●●●●●●

Retype password ●●●●●●●●●

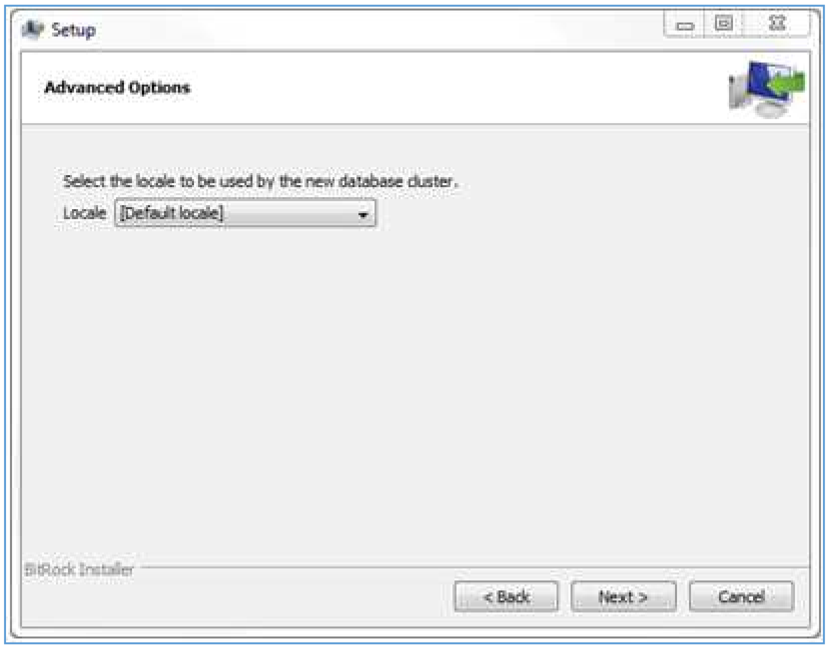
- Enter the port for PostgreSQL. Make sure that no other applications are using this port. Postgres by default uses port 5432.



Please select the port number the server should listen on.

Port 5432

- Choose the default locale, this by default would select UTF-8 support for the datasbase.



Select the locale to be used by the new database cluster.

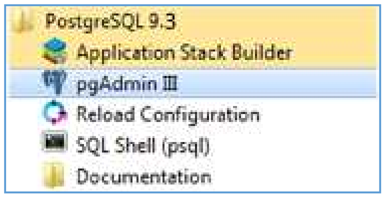
Locale [Default locale]

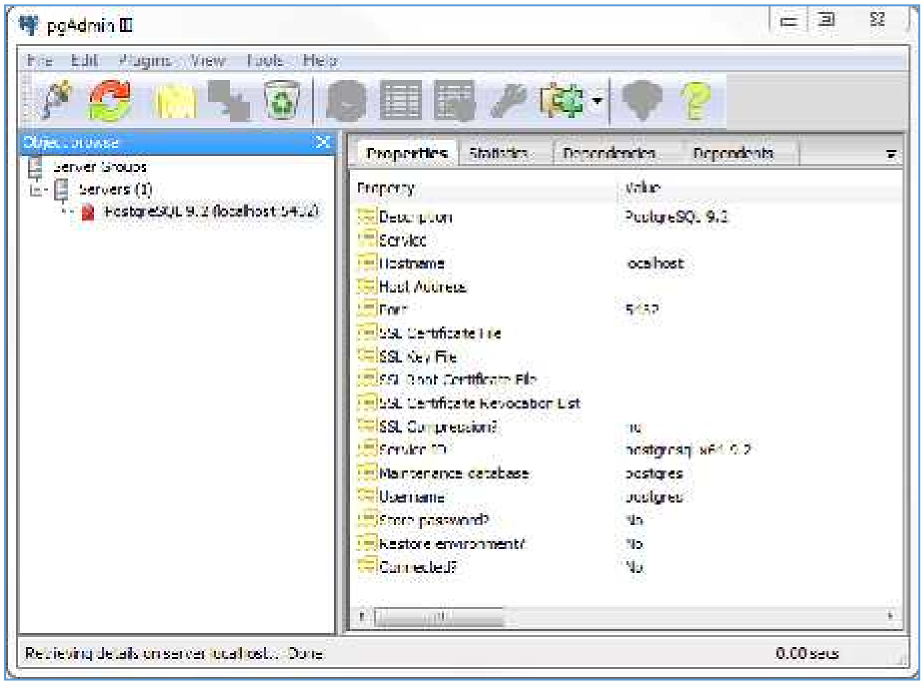
Verifying the Installation

There are several ways to verify the installation. You can try to connect to the PostgreSQL database server from any client application e.g., psql, pgAdmin, etc.

Use pgAdmin III application installed with postgresql database to check if database installed correctly and database service is running.

Click on pgAdmin III to GUI.




pgAdmin III
File Edit Plugins View Tools Help
Object browser
Server Groups
Servers (1)
PostgreSQL 9.3 (localhost:5432)
Properties Statistics Dependencies Dependents

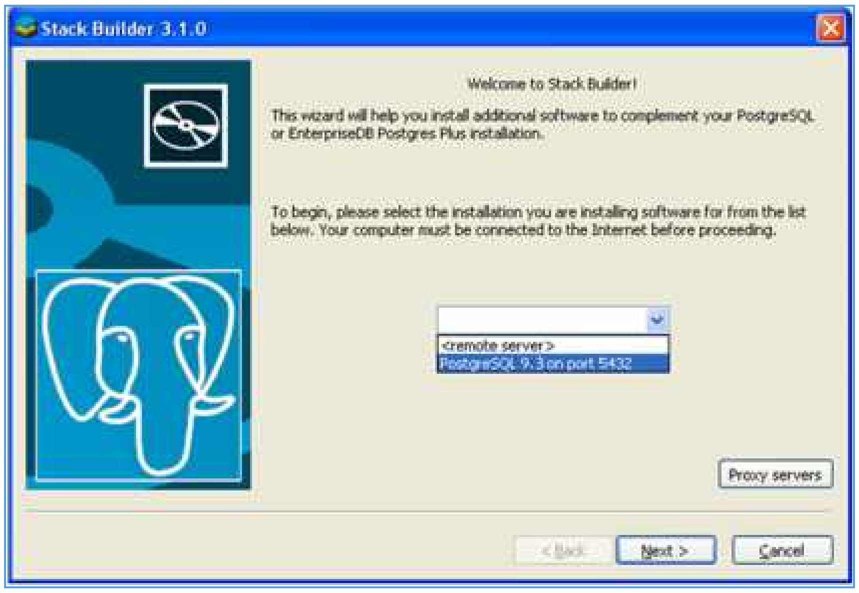
Property	Value
Description	PostgreSQL 9.3
Service	
Hostname	localhost
Host Address	
Port	5432
SSL Certificate File	
SSL Key File	
SSL Root Certificate File	
SSL Certificate Revocation List	
SSL Compression?	no
Service ID	postgresql-x64-9.3
Maintenance database	postgres
Username	postgres
Store password?	No
Restore environment?	No
Connected?	No

Retrieving details on server localhost... Done 0.00 secs

Double click on **PostgreSQL 9.3** on the object browser. It will ask you for admin password for postgres user. Enter the password used in the installation step. Successful login will display all the objects belonging to the server.

2.1.4 Installing PostGIS

After Postgres Installation is complete the installer will launch the Stack Builder application for installing Postgres extensions.



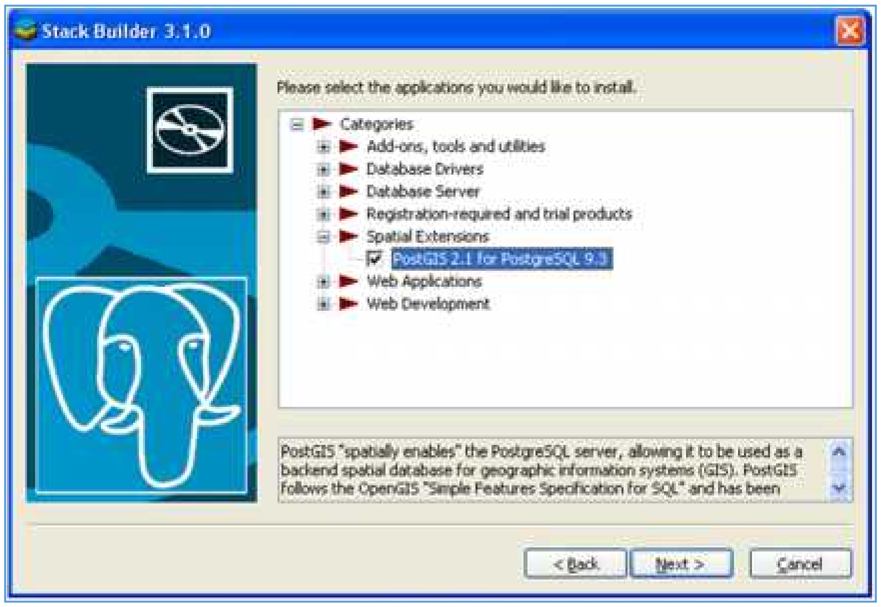
Welcome to Stack Builder!

This wizard will help you install additional software to complement your PostgreSQL or EnterpriseDB Postgres Plus installation.

To begin, please select the installation you are installing software for from the list below. Your computer must be connected to the Internet before proceeding.

<remote server>
PostgreSQL 9.3 on port 5432

Select the postgres database that has been installed, from the dropdown.



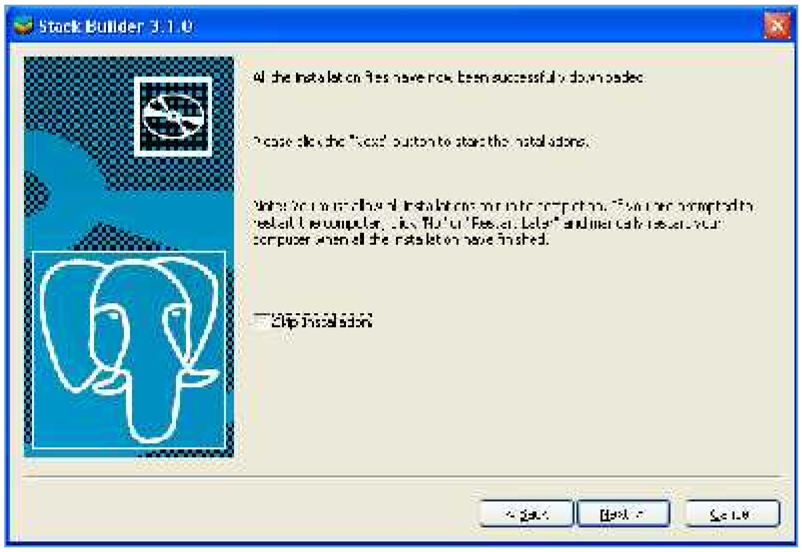
Please select the applications you would like to install.

- Categories
  - Add-ons, tools and utilities
  - Database Drivers
  - Database Server
  - Registration-required and trial products
  - Spatial Extensions
    - ☑ PostGIS 2.1 for PostgreSQL 9.3
  - Web Applications
  - Web Development

PostGIS "spatially enables" the PostgreSQL server, allowing it to be used as a backend spatial database for geographic information systems (GIS). PostGIS follows the OpenGIS "Simple Features Specification for SQL" and has been

Expand the spatial extension node and select the postGIS option. Click Next.

Click on the next few option to download the postGIS extension.


Stack Builder 3.1.0

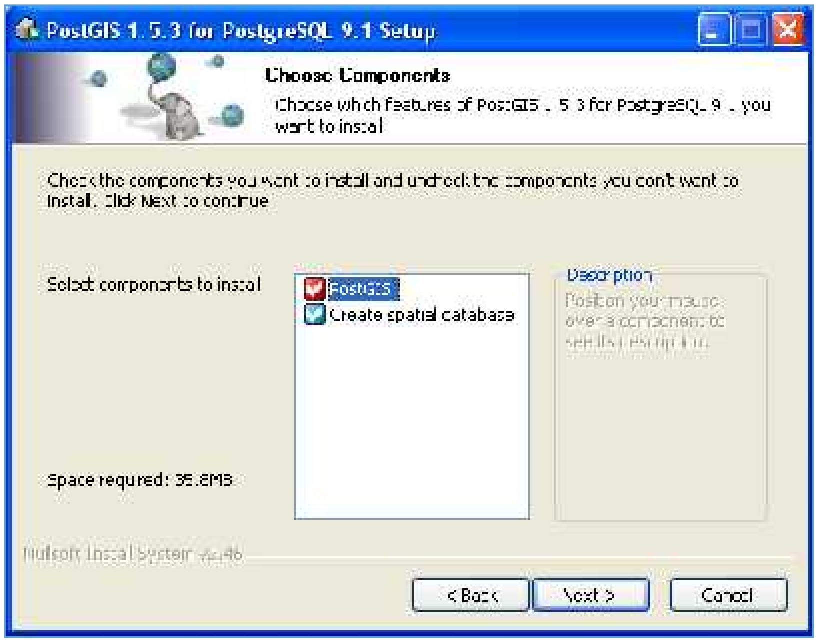
All the installation files have now been successfully downloaded.

Please click the "Next" button to start the installations.

Note: You must allow all installations to run to completion. If you are prompted to restart the computer, click "No" or "Restart Later" and manually restart your computer when all the installations have finished.

☐ Skip Installation

Click on Next button to start PostGIS installation.


PostGIS 1.5.3 for PostgreSQL 9.1 Setup

Choose Components

Choose which features of PostGIS 1.5.3 for PostgreSQL 9.1 you want to install.

Check the components you want to install and uncheck the components you don't want to install. Click Next to continue.

Select components to install:

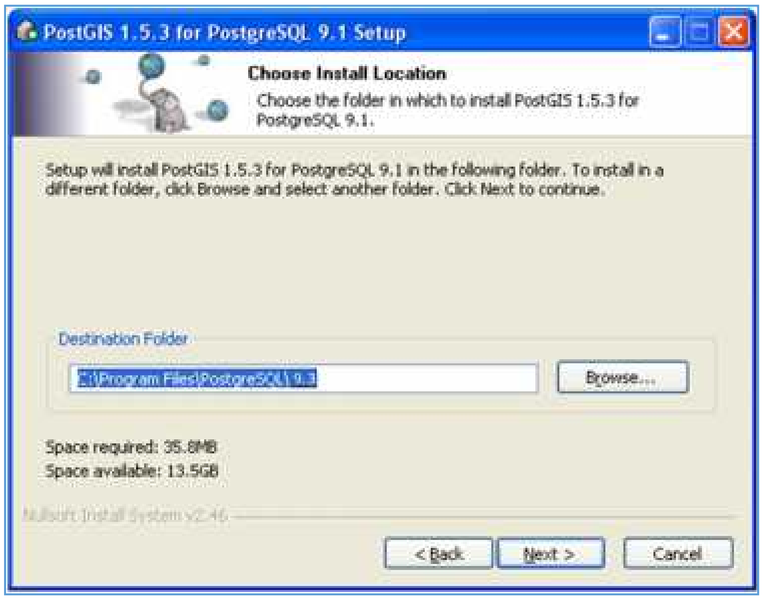
- [✓] PostGIS
- [✓] Create spatial database

Description

Position your mouse over a component to see its description.

Space required: 35.8MB

Use default options to install PostGIS as well as create a Spatial Database.


PostGIS 1.5.3 for PostgreSQL 9.1 Setup
Browse...
< Back
Next >
Cancel
Nullsoft Install System v2.46

Choose Install Location

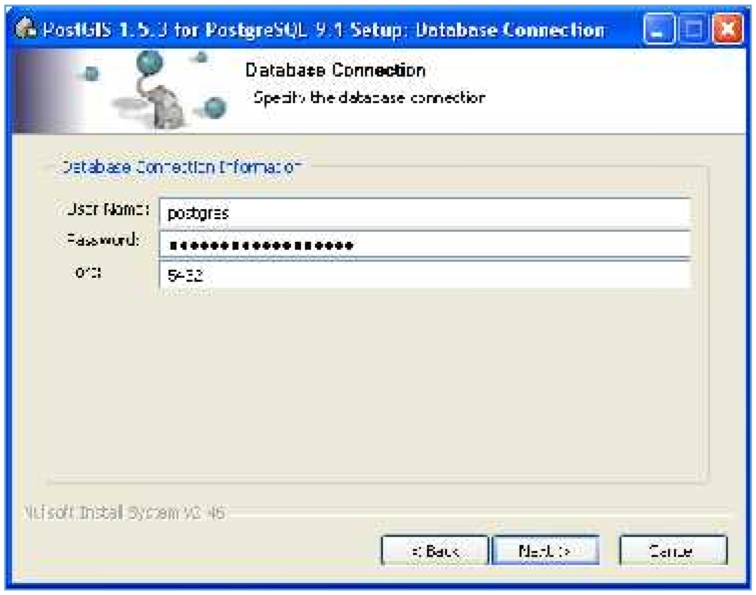
Choose the folder in which to install PostGIS 1.5.3 for PostgreSQL 9.1.

Setup will install PostGIS 1.5.3 for PostgreSQL 9.1 in the following folder. To install in a different folder, click Browse and select another folder. Click Next to continue.

Destination Folder: C:\Program Files\PostgreSQL\9.1

Space required: 35.8MB
Space available: 13.5GB

Choose Location of PostGIS extension installation.


PostGIS 1.5.3 for PostgreSQL 9.1 Setup: Database Connection
< Back
Next >
Cancel
Nullsoft Install System v2.46

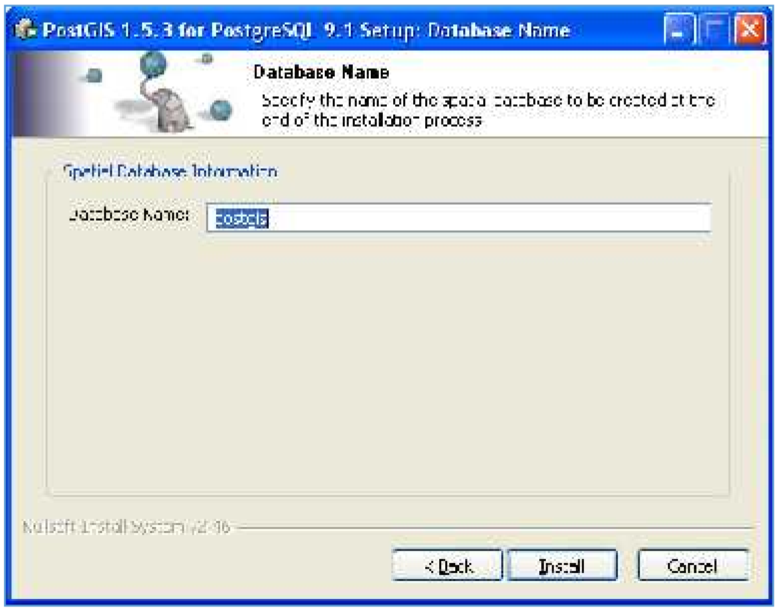
Database Connection

Specify the database connection

Database Connection Information

User Name: postgres
Password: ••••••••••••••••••
Port: 5432

Provide the credentials of the postgresql database over which PostGIS is to be installed. This would be the same as provided while installing Postgres database.


PostGIS 1.5.3 for PostgreSQL 9.1 Setup: Database Name

Database Name

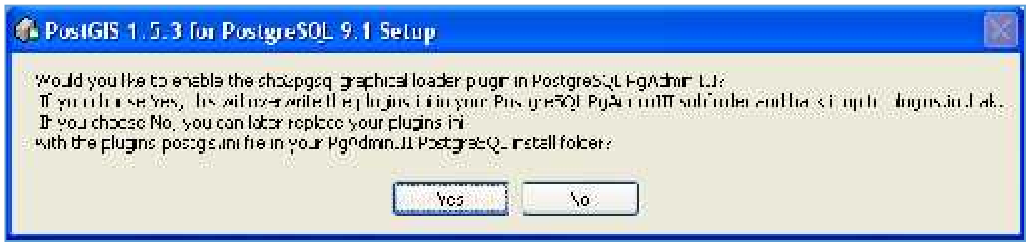
Specify the name of the spatial database to be created at the end of the installation process

Spatial Database Information

Database Name: postgis

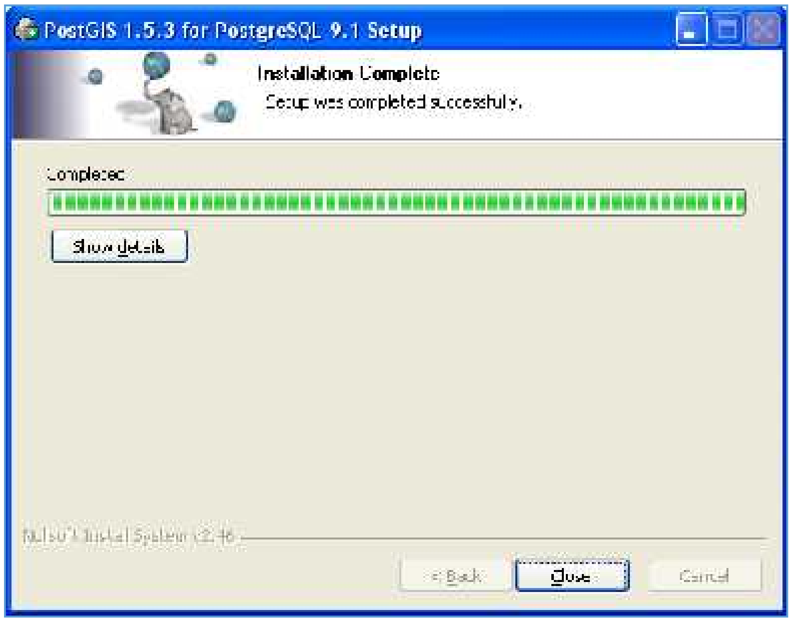
Nullsoft Install System v2.46

Keep the spatial database Name as postgis (default).


PostGIS 1.5.3 for PostgreSQL 9.1 Setup

Would you like to enable the shp2pgsql graphical loader plugin in PostgreSQL PgAdmin III?
If you choose Yes, this will overwrite the plugins.ini in your PostgreSQL PgAdmin III subfolder and back it up to plugins.ini.bak.
If you choose No, you can later replace your plugins.ini
with the plugins.postgis.ini file in your PgAdmin III PostgreSQL install folder?

Click on Yes to install shapefile loader.


PostGIS 1.5.3 for PostgreSQL 9.1 Setup

Installation Complete

Setup was completed successfully.

Completed

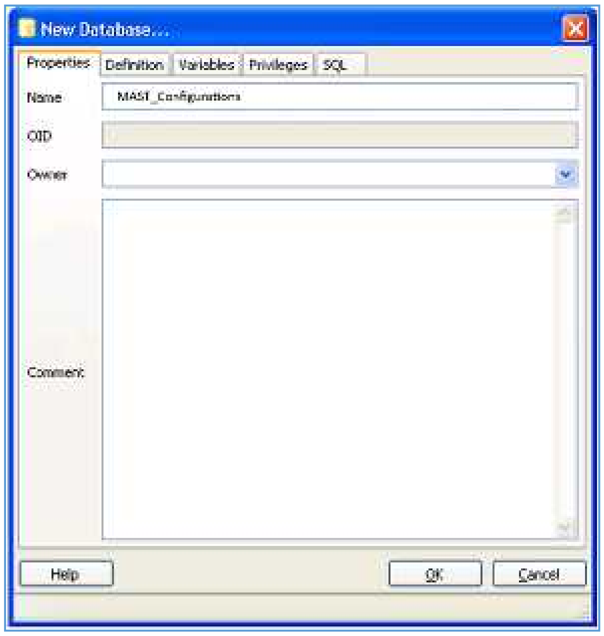
Nullsoft Install System v2.46

postGIS installation complete.

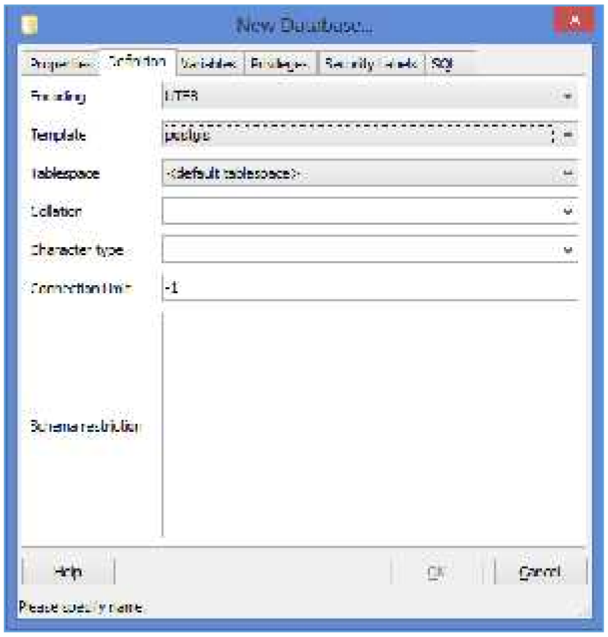
2.1.5 Creating and populating database from script

Below steps defines the steps involved in restoring the database backup file downloaded from GITHub.

- Download the postgresql database scripts from GITHub (https://github.com/MASTUSAID/DB-SCRIPTS)
- Copy the database scripts file in the local system/server .
- Open pgAdminIII.
- Connect to postgres database using the authentication specified while installing the database.



Name: MAST_Configurations
OID:
Owner:
Comment:



Encoding: UTF8
Template: postgis
Tablespace: <default tablespace>
Collation:
Character type:
Connection limit: -1
Schema restriction:

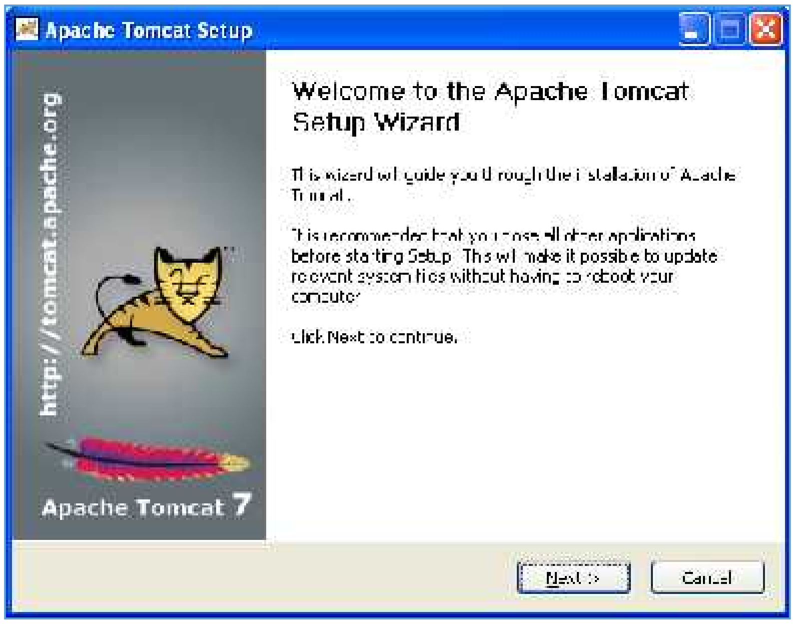
Please specify name.

- Right click on Database node and Click on NEW DATABASE option. Provide a database name (MAST_Configurations) it should match with the database bak file that you are going to restore.
- Goto definition tab and select template as postgis from the dropdown. This would enable spatial support to the newly created database.
- Click OK. New database would be created.
- Run the script mast_configurations_schema.sql to create database schema.
- Once database schema is created run the script mast_configurations_data.sql to populate master tables.

- Validate whether all the tables have been created and master data populated in this new database.

2.1.6 Apache Tomcat Installation

Download or use the latest version of Apache Tomcat Server installer executable file.



Apache Tomcat Setup

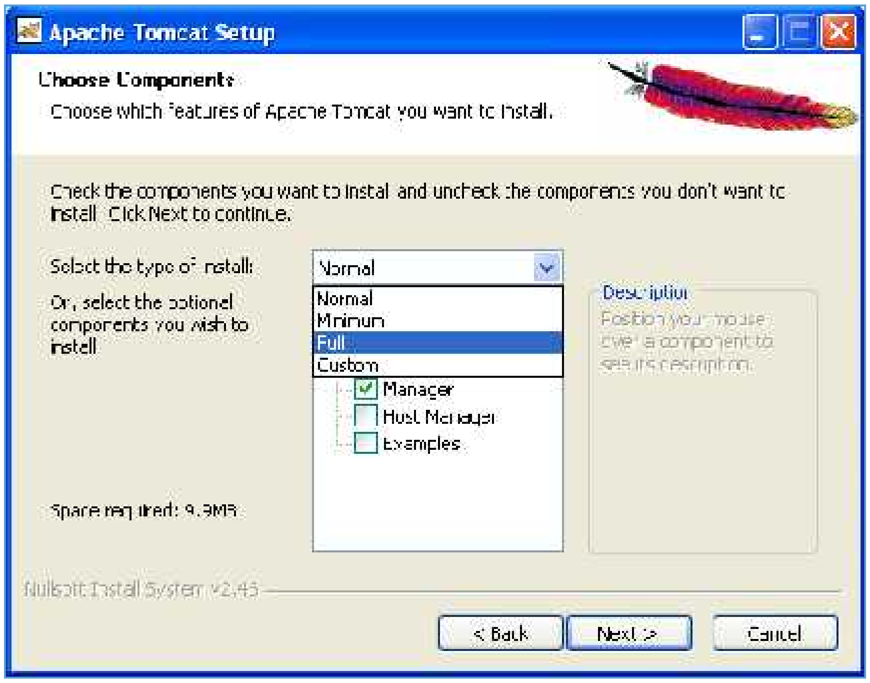
Welcome to the Apache Tomcat Setup Wizard

This wizard will guide you through the installation of Apache Tomcat.

It is recommended that you close all other applications before starting Setup. This will make it possible to update relevant system files without having to reboot your computer.

Click Next to continue.

Click Next.



Apache Tomcat Setup

Choose Components

Choose which features of Apache Tomcat you want to install.

Check the components you want to install and uncheck the components you don't want to install. Click Next to continue.

Select the type of install: Normal

Or, select the optional components you wish to install:

Normal
Minimum
Full
Custom

- [✓] Manager
- [ ] Host Manager
- [ ] Examples

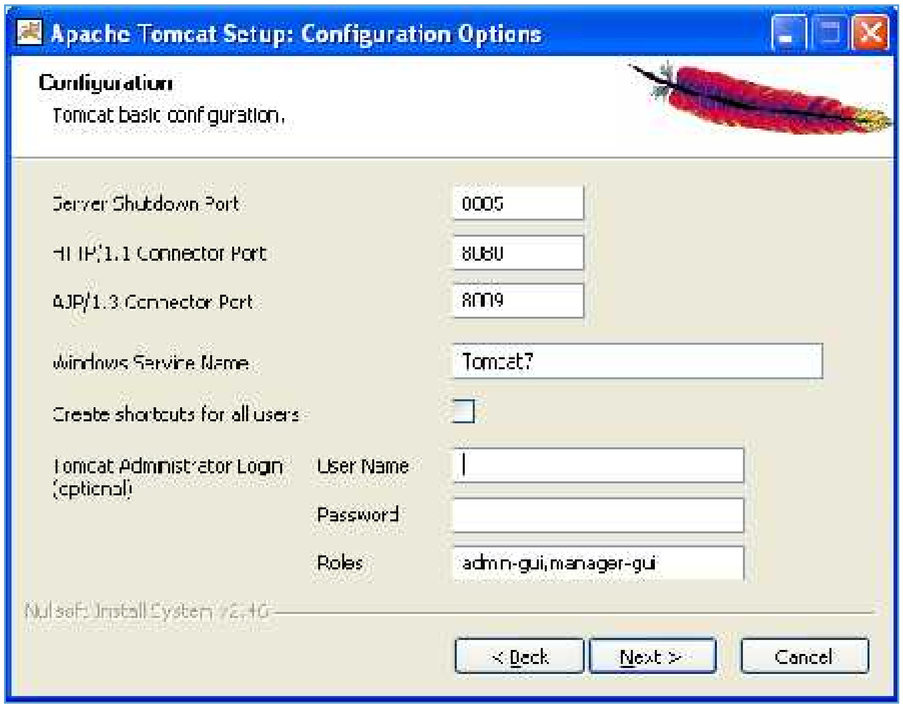
Description

Position your mouse over a component to see its description.

Space required: 9.9MB

Nullsoft Install System v2.45

Choose Type of Install as FULL to install all features.



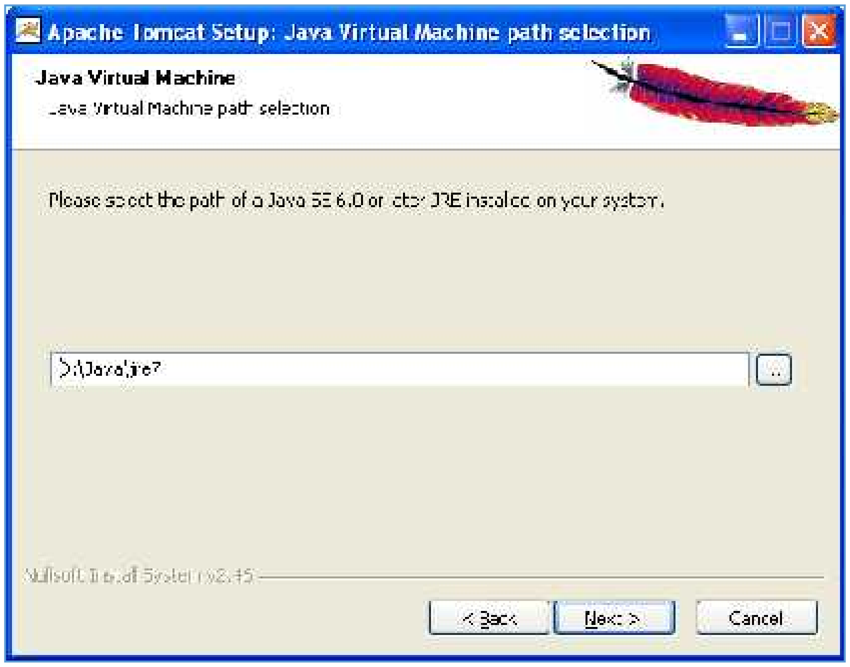
Apache Tomcat Setup: Configuration Options

Configuration
Tomcat basic configuration.

Server Shutdown Port		8005
HTTP/1.1 Connector Port		8080
AJP/1.3 Connector Port		8009
Windows Service Name		Tomcat7
Create shortcuts for all users		☐
Tomcat Administrator Login (optional)	User Name	
	Password	
	Roles	admin-gui,manager-gui

Nullsoft Install System v2.46

In configuration option keep everything as default. Provide the user name and password and note it somewhere for future reference.



Apache Tomcat Setup: Java Virtual Machine path selection

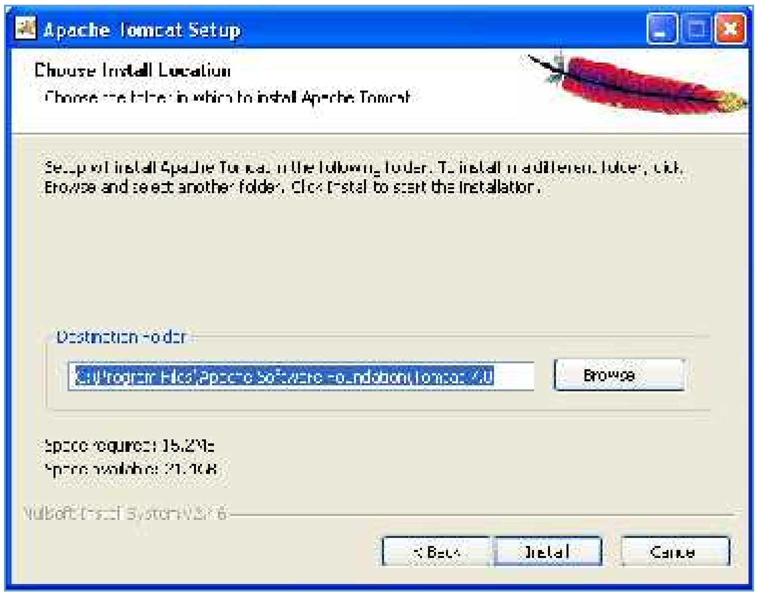
Java Virtual Machine
Java Virtual Machine path selection.

Please select the path of a Java SE 6.0 or later JRE installed on your system.

D:\Java\jre7

Nullsoft Install System v2.46

Select the path of JRE as installed in the local system/server.


Apache Tomcat Setup

Choose Install Location

Choose the folder in which to install Apache Tomcat.

Setup will install Apache Tomcat in the following folder. To install in a different folder, click Browse and select another folder. Click Install to start the installation.

Destination Folder: C:\Program Files\Apache Software Foundation\Tomcat 7.0

Space required: 15.2MB
Space available: 21.1GB

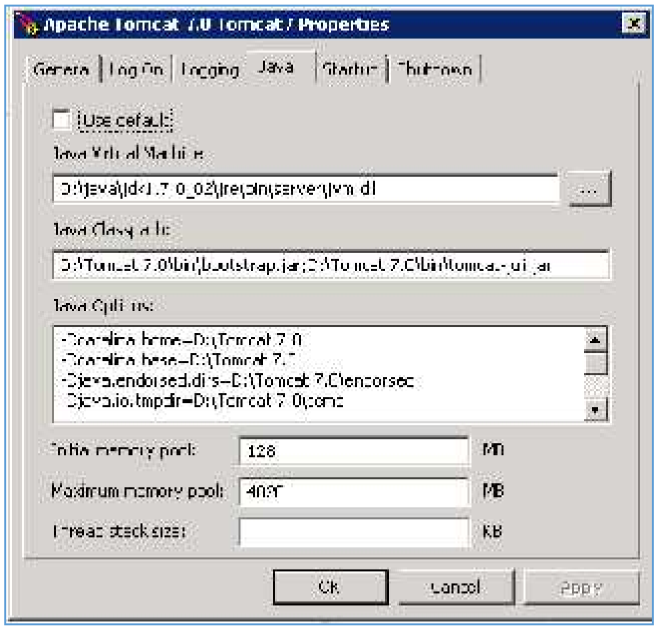
Provide installation path of Tomcat and click on install. This would install the Tomcat Server as a Service in the local system/ Server.

2.1.7 Apache Tomcat Configuration

Below steps define configuration of Apache tomcat application server to be used with development environment to deploy the application in debug/run mode.

Memory Allocation

Go to Tomcat Home/BIN folder right click on Tomcat7w.exe file and click on Run as Administrator.


Apache Tomcat 7.0 Tomcat7 Properties

Use default: [ ]

Java Virtual Machine: D:\java\jdk1.7.0_02\jre\bin\server\jvm.dll

Java Classpath: D:\Tomcat 7.0\bin\bootstrap.jar;D:\Tomcat 7.0\bin\tomcat-juli.jar

Java Options:

```
-Dcatalina.home=D:\Tomcat 7.0
-Dcatalina.base=D:\Tomcat 7.0
-Djava.endorsed.dirs=D:\Tomcat 7.0\endorsed
-Djava.io.tmpdir=D:\Tomcat 7.0\temp
```

Initial memory pool: 128 MB

Maximum memory pool: 4096 MB

Thread stack size: KB

Apache Tomcat Properties window will popup. Click on Java Tab.

Add the following line at the end in Java Options Textbox.

“-XX:MaxPermSize=256m”

Set the Initial memory pool to 128 and maximum memory pool to 4028.

Click OK.

Configuring Context .xml and Server.xml

Context.xml : Add the line marked in RED font to context.xml file (Blue colored font describes the content of default context.xml file).

```
<?xml version='1.0' encoding='utf-8'?>
<!-- The contents of this file will be loaded for each web application -->
<Context>
   <!-- Default set of monitored resources -->
   <WatchedResource>WEB-INF/web.xml</WatchedResource>

   <!-- Uncomment this to disable session persistence across Tomcat restarts -->
   <!-- Uncomment this to enable Comet connection tacking (provides events
       on session expiration as well as webapp lifecycle) -->
   <!-- <Valve className="org.apache.catalina.valves.CometConnectionManagerValve" />
   -->
   <ResourceLink global="jdbc/dbname" name="jdbc/dbname"
type="javax.sql.DataSource"/>
</Context>
```

Server.xml: Edit the server.xml file. Add the lines as shown in red font in the server.xml file. Change the database/user and password of the postgres database to be configured with Spatialvue (as provided during postgres installation).

Please note that these changes are to be done only in the lines (shown in red font below) that are to be added in server.xml.

```
<?xml version='1.0' encoding='utf-8'?>
<Server port="8005" shutdown="SHUTDOWN">
  <!-- Security listener. Documentation at /docs/config/listeners.html
  <Listener className="org.apache.catalina.security.SecurityListener" />
  -->
  <!--APR library loader. Documentation at /docs/apr.html -->
  <Listener className="org.apache.catalina.core.AprLifecycleListener" SSLEngine="on" />
  <!--Initialize Jasper prior to webapps are loaded. Documentation at /docs/jasper-howto.html -->
  <Listener className="org.apache.catalina.core.JasperListener" />
  <!-- Prevent memory leaks due to use of particular java/javax APIs-->
  <Listener className="org.apache.catalina.core.JreMemoryLeakPreventionListener" />
  <Listener className="org.apache.catalina.mbeans.GlobalResourcesLifecycleListener" />
  <Listener className="org.apache.catalina.core.ThreadLocalLeakPreventionListener" />
  <!-- Global JNDI resources
       Documentation at /docs/jndi-resources-howto.html
  -->
  <GlobalNamingResources>
    <!-- Editable user database that can also be used by
         UserDatabaseRealm to authenticate users
    -->
    <Resource name="UserDatabase" auth="Container"
              type="org.apache.catalina.UserDatabase"
              description="User database that can be updated and saved"
              factory="org.apache.catalina.users.MemoryUserDatabaseFactory"
              pathname="conf/tomcat-users.xml" />
    <Resource type="javax.sql.DataSource"
            name="jdbc/dbname"
            factory="org.apache.tomcat.jdbc.pool.DataSourceFactory"
            driverClassName="org.postgresql.Driver"
            url="jdbc:postgresql://host:5432/dbname"
            username="username"
            password="password"
                                    testWhileIdle="true"
              testOnBorrow="true"
              testOnReturn="false"
              validationQuery="SELECT 1"
              validationInterval="30000"
              timeBetweenEvictionRunsMillis="5000"
              maxActive="100"
              minIdle="10"
              maxWait="10000"
              initialSize="10"
              removeAbandonedTimeout="60"
              removeAbandoned="true"
              logAbandoned="true"
              minEvictableIdleTimeMillis="30000"
              jmxEnabled="true"
              jdbcInterceptors="org.apache.tomcat.jdbc.pool.interceptor.ConnectionState;
                   org.apache.tomcat.jdbc.pool.interceptor.StatementFinalizer;

org.apache.tomcat.jdbc.pool.interceptor.SlowQueryReportJmx(threshold=10000)"
            />
  </GlobalNamingResources>

  <!-- A "Service" is a collection of one or more "Connectors" that share
       a single "Container" Note:  A "Service" is not itself a "Container",
       so you may not define subcomponents such as "Valves" at this level.
       Documentation at /docs/config/service.html
```

```
   -->
  <Service name="Catalina">

    <!--The connectors can use a shared executor, you can define one or more named thread
pools-->
    <!--
    <Executor name="tomcatThreadPool" namePrefix="catalina-exec-"
        maxThreads="150" minSpareThreads="4"/>
    -->


    <!-- A "Connector" represents an endpoint by which requests are received
         and responses are returned. Documentation at :
         Java HTTP Connector: /docs/config/http.html (blocking & non-blocking)
         Java AJP  Connector: /docs/config/ajp.html
         APR (HTTP/AJP) Connector: /docs/apr.html
         Define a non-SSL HTTP/1.1 Connector on port 8080
    -->
    <Connector port="8080" protocol="HTTP/1.1"
               connectionTimeout="20000"
               redirectPort="8443" />
    <!-- A "Connector" using the shared thread pool-->
    <!--
    <Connector executor="tomcatThreadPool"
               port="8080" protocol="HTTP/1.1"
               connectionTimeout="20000"
               redirectPort="8443" />
    -->
    <!-- Define a SSL HTTP/1.1 Connector on port 8443
         This connector uses the JSSE configuration, when using APR, the
         connector should be using the OpenSSL style configuration
         described in the APR documentation -->
    <!--
    <Connector port="8443" protocol="HTTP/1.1" SSLEnabled="true"
               maxThreads="150" scheme="https" secure="true"
               clientAuth="false" sslProtocol="TLS" />
    -->

    <!-- Define an AJP 1.3 Connector on port 8009 -->
    <Connector port="8009" protocol="AJP/1.3" redirectPort="8443" />


    <!-- An Engine represents the entry point (within Catalina) that processes
         every request.  The Engine implementation for Tomcat stand alone
         analyzes the HTTP headers included with the request, and passes them
         on to the appropriate Host (virtual host).
        Documentation at /docs/config/engine.html -->

    <!-- You should set jvmRoute to support load-balancing via AJP ie :
    <Engine name="Catalina" defaultHost="localhost" jvmRoute="jvm1">
    -->
    <Engine name="Catalina" defaultHost="localhost">

      <!--For clustering, please take a look at documentation at:
          /docs/cluster-howto.html  (simple how to)
          /docs/config/cluster.html (reference documentation) -->
      <!--
      <Cluster className="org.apache.catalina.ha.tcp.SimpleTcpCluster"/>
      -->

      <!-- Use the LockOutRealm to prevent attempts to guess user passwords
           via a brute-force attack -->
      <Realm className="org.apache.catalina.realm.LockOutRealm">
        <!-- This Realm uses the UserDatabase configured in the global JNDI
             resources under the key "UserDatabase".  Any edits
             that are performed against this UserDatabase are immediately
             available for use by the Realm.  -->
        <Realm className="org.apache.catalina.realm.UserDatabaseRealm"
               resourceName="UserDatabase"/>
      </Realm>
      <Host name="localhost"  appBase="webapps"
            unpackWARs="true" autoDeploy="true">
        <!-- SingleSignOn valve, share authentication between web applications
             Documentation at: /docs/config/valve.html -->
        <!--
        <Valve className="org.apache.catalina.authenticator.SingleSignOn" />
        -->

        <!-- Access log processes all example.
             Documentation at: /docs/config/valve.html
             Note: The pattern used is equivalent to using pattern="common" -->
        <Valve className="org.apache.catalina.valves.AccessLogValve" directory="logs"
               prefix="localhost_access_log." suffix=".txt"
               pattern="%h %l %u %t &quot;%r&quot; %s %b" />

      </Host>
    </Engine>
  </Service>
</Server>
```

2.1.8 Eclipse Luna

Eclipse luna Integrated Development Environment is used as development platform for coding and compiling of MAST web infrastructure source code downloaded from GITHub.

Download and install Eclipse Luna in local machine.

Install Maven plugin dependency. Maven plugin can be installed from Eclipse market place.

2.2 MAST Mobile Data Capture Application (Android)

The core tools and software used to develop MAST mobile application are:

- Java JDK7
- Eclipse Luna IDE
- Android Development Tools (ADT Plugin)
- Android Software Development Kit (SDK)

2.2.1 Installing Java JDK 7

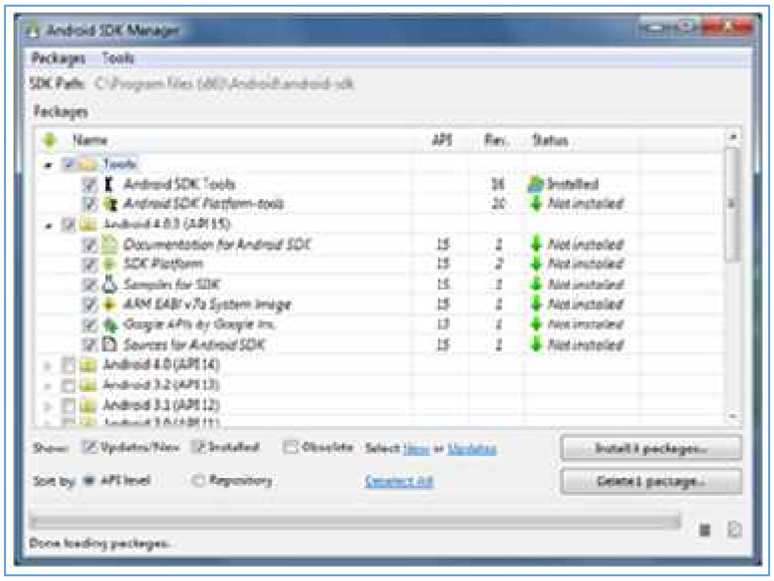
Installation method is same as defined in section 2.1.1.

2.2.2 Installing the Stand-alone SDK Tools

Download the SDK from the following url: https://developer.android.com/sdk/index.html#Other

The downloaded package is an executable file that starts an installer. The installer checks your machine for required tools and installs it if necessary.

Double-click the executable (.exe file) to start the install.



Android SDK Manager

Packages Tools

SDK Path: C:\Program Files (x86)\Android\android-sdk

Packages

Name	API	Rev.	Status
Tools			
Android SDK Tools		16	Installed
Android SDK Platform-tools		10	Not installed
Android 4.0.3 (API 15)			
Documentation for Android SDK	15	1	Not installed
SDK Platform	15	2	Not installed
Samples for SDK	15	1	Not installed
ARM EABI v7a System Image	15	1	Not installed
Google APIs by Google Inc.	15	1	Not installed
Sources for Android SDK	15	1	Not installed
Android 4.0 (API 14)			
Android 3.2 (API 13)			
Android 3.1 (API 12)			
Android 3.0 (API 11)			

Show: [✓] Updates/New [✓] Installed [ ] Obsolete Select New or Updates

Sort by: (●) API level ( ) Repository Deselect All

Done loading packages.

Once the installation completes, the installer starts the Android SDK Manager.

As a minimum when setting up the Android SDK, you should download the latest tools and Android platform:

Open the Tools directory and select:

- Android SDK Tools
- Android SDK Platform-tools
- Android SDK Build-tools (highest version)

Open the first Android X.X folder (the latest version) and select:

- SDK Platform
- A system image for the emulator, such as ARM EABI v7a System Image.

For using Google Maps API download Google Play services package. Open the Extras directory and select:

- Google Repository
- Google Play services

Once you've selected all the desired packages, continue to install:

- Click Install X packages.
- In the next window, double-click each package name on the left to accept the license agreement for each.
- Click Install.

2.2.3 Installing the Eclipse Plugin

To add the ADT plugin to Eclipse:

- Start Eclipse, then select **Help** > **Install New Software**.
- Click **Add**, in the top-right corner.
- In the Add Repository dialog that appears, enter "ADT Plugin" for the *Name* and the following URL for the *Location*: https://dl-ssl.google.com/android/eclipse/

Note: The Android Developer Tools update site requires a secure connection. Make sure the update site URL you enter starts with HTTPS.

- Click **OK**.
- In the Available Software dialog, select the checkbox next to Developer Tools and click **Next**.
- In the next window, you'll see a list of the tools to be downloaded. Click **Next**.
- Read and accept the license agreements, then click **Finish**.

If you get a security warning saying that the authenticity or validity of the software can't be established, click **OK**.

- When the installation completes, restart Eclipse.

Once Eclipse restarts, you must specify the location of your Android SDK directory:

- In the "Welcome to Android Development" window that appears, select Use existing SDKs.
- Browse and select the location of the Android SDK directory you recently downloaded.
- Click Next.

Your Eclipse IDE is now set up to develop Android apps, but you need to add the latest SDK platform tools and an Android platform to your environment.

3 DOWNLOAD AND DEPLOY SOURCE CODE

3.1 Download & Deploy Source Code (Web Infrastructure)

Visit https://github.com/MASTUSAID/MAST-DMI to download MAST Data Management Infrastructure Source code. Download the zip file on to local system by clicking Download ZIP button.

3.1.1 Configure and Build Source Code

Prerequisite: For configuring the source code in Eclipse, it is mandatory to have working knowledge of Eclipse IDE and Maven. Without knowledge of these it would be difficult for user to setup the source code and configure jdk environment to work in debug environment.

Extract the downloaded zip file to local computer. Open Eclipse and create a new workspace in the folder where you extracted the source code.

On the Project Explorer/Package Explorer right click and select import→ import…

Expand Maven and select" Existing Maven Projects". Press next. Select the root folder of extracted MAST archive where pom.xml is located. Click finish.

Above activity would set up the maven project on to the new eclipse workspace that you created.

Set the Tomcat/webapp path for locally installed tomcat. This path can be set in pom.xml under artifact **maven-war-plugin.**

```
<plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-war-plugin</artifactId>
        <version>2.1.1</version>
        <configuration>
                <warName>mast</warName>
                <outputDirectory>E:\apache-tomcat-7.0.56\webapps</outputDirectory>
                <packagingExcludes>WEB-INF/web.xml,WEB-INF/lib/org.apache.xerces-
2.9.0.jar,WEB-INF/lib/org.apache.xerces-2.9.0-SNAPSHOT.jar</packagingExcludes>
        </configuration>
</plugin>
```

From project menu click on clean option. Select project MAST and press OK. Ensure build automatically option is checked before running the clean.

Generating war file- For creating war file, on project explorer right click on the project root and select Run→Maven Install. This option would initially download all the maven dependencies as configured in pom.xml and then create war file in the tomcat/webapp path as configured in pom.xml.

3.1.2 Install Geoserver

Copy the geoserver.war to Tomcat/Webapps folder.

From Run Command type "Services.msc" to open services window. From the list of services select the **Apache Tomcat 7.0** service stop and restart the service.

When the service starts tomcat would automatically extract the war file in webapp folder.

3.2 Download & Deploy Source Code (Mobile Data Capture Application)

3.2.1 Download Source code

Visit https://github.com/MASTUSAID/MAST-MOBILE to download MAST Mobile Application source code. Download the zip file on to local system by clicking Download ZIP button.

3.2.2 Configure and Build Source code

Prerequisite: For configuring the source code in Eclipse, it is mandatory to have working knowledge of Eclipse IDE and Android SDK. Without knowledge of these it would be difficult for the user to setup the source code and configure to work in debug environment.

Extract the downloaded zip file to local computer. Open Eclipse and create a new workspace in the folder where you extracted the source code.

On the Project Explorer/Package Explorer right click and select import → import. Choose Existing Android code into workspace option and browse to the project root folder and click Finish.

Above activity would set up the android project on to the new eclipse workspace that you just created.