

Distributed Polling System Installation Manual

Version 1.0

Table of Contents

1. Introduction	3
1.1 Purpose of this document	3
1.2 Intended Audience	3
1.3 Scope	3
1.4 Definitions and acronyms	3
1.4.1 Acronyms and abbreviations	3
2. System Requirements	3
2.1 Hardware Requirements	3
2.2 Software Requirements	3
2.3 Operating Systems	3
3. DPS Installation	4
3.1 Installing the mobile communication sub layer	4
3.1.1 Installing Java Communication API	4
3.1.2 Installing Device Drivers	4
3.1.3 Terminal Program	5
3.1.3.1 Configuring Connection with HyperTerminal (HT)	5
3.1.3.2 Test Connection Using HyperTerminal	7
3.1.3.3 Checking AT Commands Support Using HyperTerminal	8
3.1.3.4 Finding the SMSC Address	9
3.2 Installing the email communication sub layer	10
3.2.1 Email Server Installation and Configuration	10
3.2.2 Email Client Software Installation	10
3.3 Installing the middleware	11
3.3.1 Installing Jboss ESB server for windows	11
3.3.2 Compiling and Deployment to ESB server	12
3.4 Installing the web layer	12
3.4.1 Installing Apache Tomcat	12
3.4.2 Installing MySQL	12
3.4.3 Run web portion of DPS	13

1. Introduction

1.1 Purpose of this document

The purpose of this document is to provide details for:

- Installing the third party software application required to run DPS
- Deploying DPS components and layers
- Testing certain functionality to verify the deployment of DPS (Diagnostics)

1.2 Intended Audience

Product Users and Evaluators: Anybody who wants to use the DPS product or evaluate, needs to install DPS including SCORE supervisors and evaluators, team members, steering group and Customers.

1.3 Scope

The scope of the document is limited to providing guidance for installation, deployment and diagnostics of DPS.

1.4 Definitions and acronyms

1.4.1 Acronyms and abbreviations

Acronym or Abbreviation	Definitions
DPS	Distributed Polling System

2. System Requirements

2.1 Hardware Requirements

Two mobile devices need to be connected to the computer, which host mobile communication sub-layer.

- 1) Nokia N95 (sending SMS)
- 2) Sony Ericsson z550i (receiving SMS)

2.2 Software Requirements

The below mentioned software or their *higher versions*

- 1) JDK 1.5
- 2) Jbossesb-server-4.4.GA
- 3) apache-ant-1.7.1
- 4) WINMAIL v4.6
- 5) Eudora mail client v7.1
- 6) Tomcat v5.5
- 7) MySQL v5.0
- 8) Java Communication API (javax.comm)
- 9) Device Driver for Mobile Nokia N96 and Sony Ericsson z550i
- 10) Terminal Program (Optional)

2.3 Operating Systems

The only limitation that underlying operating system has, is that all the above mentioned 9 mandatory software's should be available for the selected operating system. We tested our application on Windows XP and Vista. For UNIX, Linux, Sun Solaris and Mac OS flavors all above the software's might not be available.

3. DPS Installation

Due to vast scope of the DPS application we divided the installation manual into four parts. All four parts can be installed in separate computers which provide a true nature of Distributed software developed in a distributed environment.

The four parts are

- Installing the mobile communication sub layer
- Installing the email communication sub layer
- Installing the middleware
- Installing the web layer

3.1 Installing the mobile communication sub layer

3.1.1 Installing Java Communication API

In order to use SMS in a computer application, an interface between the computer and the GSM network is needed. This interface is typically a mobile device connected to a serial port on the computer. In this project the serial connection to a mobile is the interface¹.

The Comm. API is specified in the package `javax.comm`, is a part of the *java extended specification* (`javax`). Sun has only implemented `javax.comm` for Microsoft Windows and Sun's Solaris/SPARC platform. But it no longer supports Implementations for Windows and few other platforms. Implementations for other platforms are supported by the RXTx project, which provides the `javax.comm` package for Linux, MacOS X, etc [1].

Download the comm. API for the appropriate OS from <http://www.rxtx.org/> or <http://blog.jservlet.com/public/serialcomm/javaSerial.zip>. The downloaded API contains the following three files (for the Windows Kit)

- a) `comm.jar`
- b) `win32comm.dll`
- c) `jaxax.comm.properties`

For java to recognize and work with serial ports, it is required to place these files in appropriate folders (JAVA_HOME is the location of JDK directory, WIN_HOME is Windows OS installation directory)

- a) Copy `comm.jar` in
 - `JAVA_HOME` → `Lib` and
 - `JAVA_HOME` → `jre` → `lib` → `ext` directory
- b) Copy `win32comm.dll` in
 - `JAVA_HOME` → `bin` and
 - `JAVA_HOME` → `jre` → `bin` and
 - `WIN_HOME` → `System32` directory
- c) Copy `jaxax.comm.properties` in
 - `JAVA_HOME` → `Lib` and
 - `JAVA_HOME` → `jre` → `lib` directory

3.1.2 Installing Device Drivers

For DPS application to send and receive SMS, it is required that the Operating System recognize the connected mobile device and hence the Java Comm. API.

Caution → By default many operating systems provide default drivers including for serial connection devices. Thus mobiles with default drivers will work like storage devices for data transfer. In order to work properly with DPS, exact driver preferably from the manufacturer need to be installed.

If we install the PC suite available for the mobile, it includes the device driver. **Download and Install the PC suite from**

a) Nokia N95 (sending SMS)

http://nds1.nokia.com/files/support/global/phones/software/Nokia_PC_Suite_7_1_18_0_eng_web.exe

¹ <http://www.daimi.au.dk/~jones/sms/report/report.html>

b) Sony Ericsson z550i (receiving SMS)

http://www.sonyericsson.com/cws/download/1/357/877/1234788930/Sony_Ericsson_PC_Suite_4.010.00_Web_EN.exe

3.1.3 Terminal Program

Microsoft HyperTerminal is a software utility that comes free with Windows XP and previous versions of Windows operating system. Microsoft Vista does not provide HyperTerminal by default, either has to be purchased or some other terminal program can be used in place of HyperTerminal.

3.1.3.1 Configuring Connection with HyperTerminal (HT)

Connect the mobiles with a valid SIM card either using USB/Bluetooth to the computer.

To start HyperTerminal (HT) utility

- In Win XP go to Start → Program → Accessories → Communication → HyperTerminal
- In Win 98 if this utility is not found we may not have installed it. To install go to Control Panel → Add/Remove Programs → Windows Setup Tab → Communication List Box Item → Details button, to install HyperTerminal

After the utility starts, it asks for a Connection Name (as in following figure). **Key in a relevant name** for ex: “SMSConnection” and press OK button.



Figure 3-1 Specifying the Connection name

HT asks for selecting the Serial port to which the above mentioned Connection is to be created. **Select the appropriate port to which the mobile is connected.**

There can be multiple ports in a Computer that are active and will be presented in the Combo-box (figure below). In order to find which port the mobile device is connected

- In Windows 98 Go to Control Panel → Modems → Diagnostic Tab
- In Windows 2000 and XP Go to Control Panel → Phone and Modem Option → Model Tab

These will display the list of connected devices and corresponding ports.

Specify the same port of your mobile device to HT and click OK



Figure 3-2 Specifying Serial Port for Mobile Connection

HT then asks for other configuration parameters such as Bit rate, data bit, parity etc. Click OK. HT is configured and is ready to accept the AT commands.

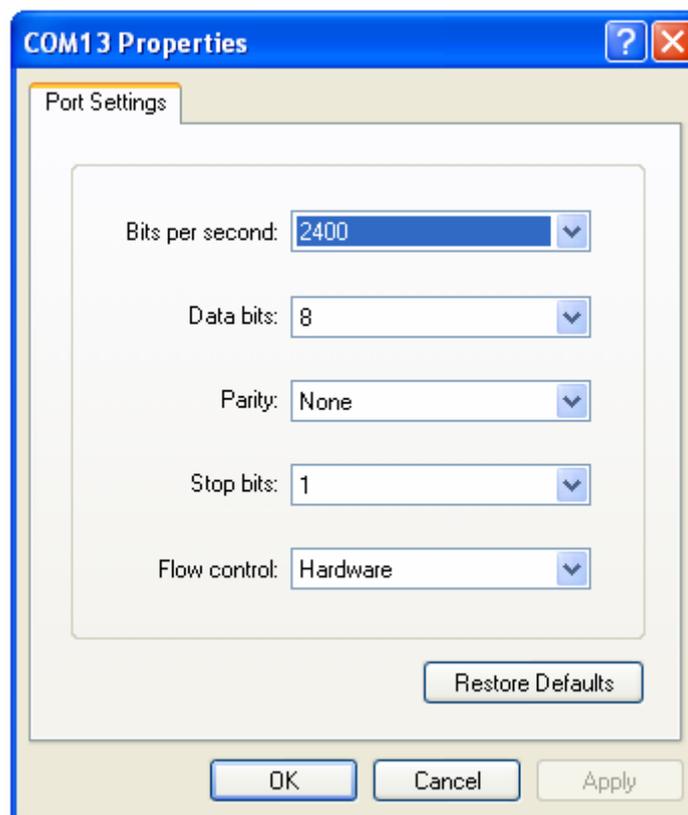


Figure 3-3 Specifying Configuration Parameters for the Connection

Usually the characters typed through keyboard are not visible. **For making the characters visible** Go to File → Properties → Setting Tab → ASCII Setup button → ASCII Sending Section → Check the Check-box, Echo typed characters locally and click OK.

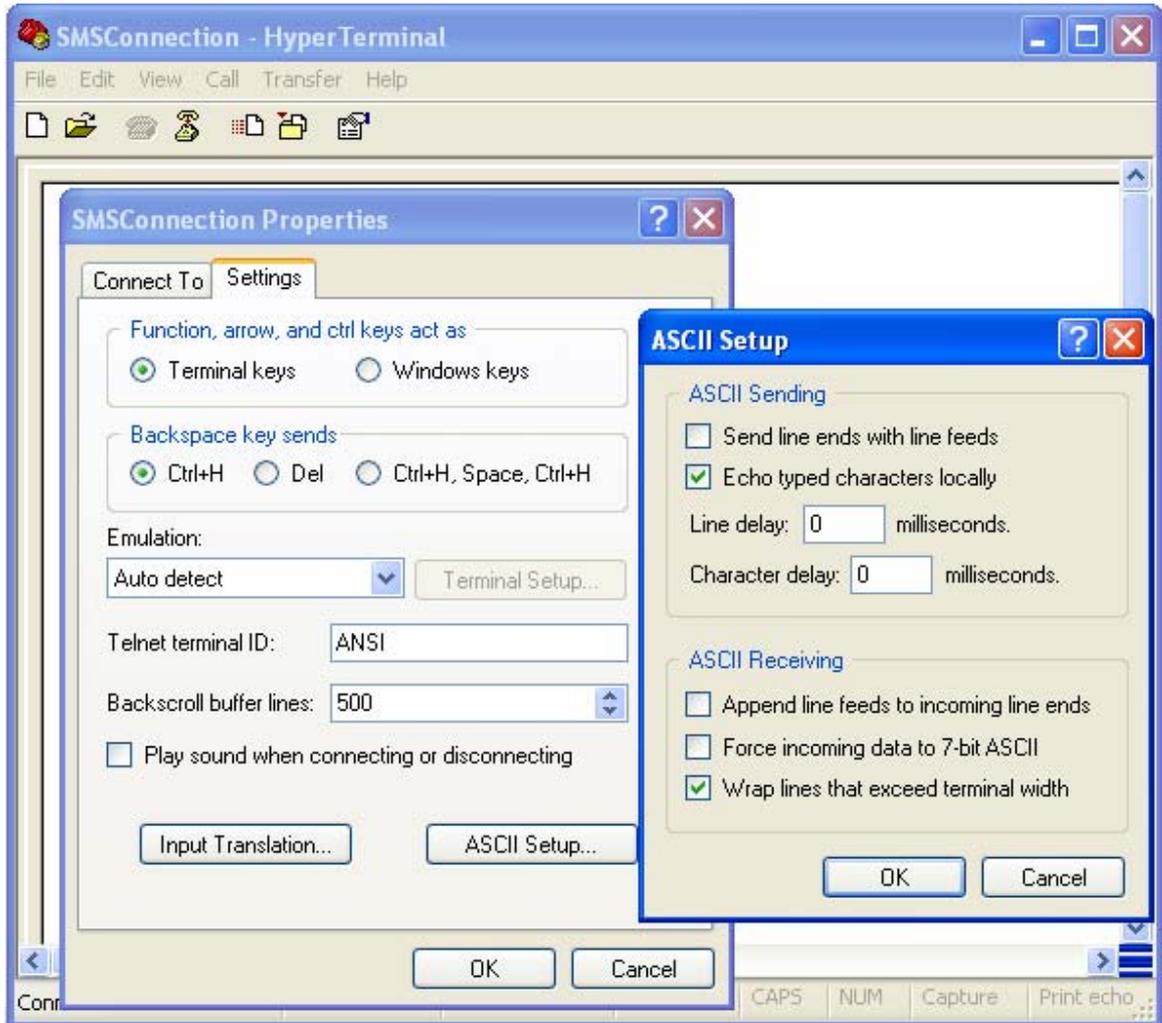


Figure 3-4 Making Screen Characters Visible

3.1.3.2. Test Connection Using HyperTerminal

For testing the connection created in the above steps, **simply type “AT” command**. This is similar to ping command in networking and which checks if the link between computer and mobile is establish and the device is responding (by sending an OK signal)

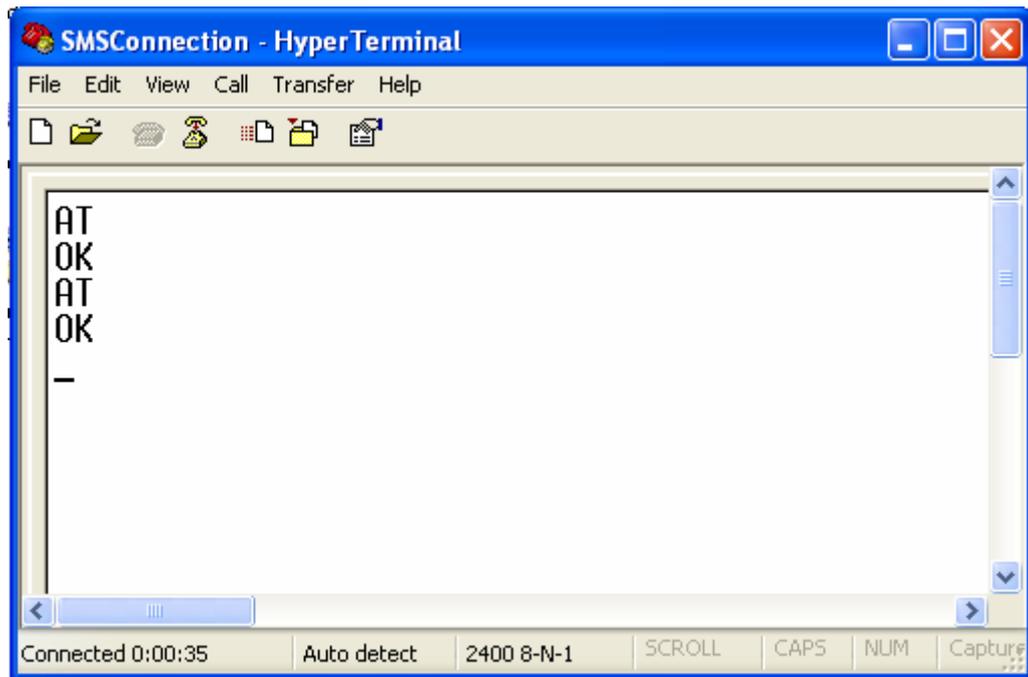


Figure 3-5 Testing Connection with Mobile device using AT Command

3.1.3.3. Checking AT Commands Support Using HyperTerminal

The mobile communication layer of DPS requires certain AT commands, to send and receive SMS. Not all mobile handset support all AT commands. The AT commands set the device supports vary with the Model of the mobile device and the Manufacturer. In order to test whether a command is supported or not use the following general syntax of

Command=?

If the result from the mobile is OK then the command “*Command*” is supported.

If the result is an “Error” or some error code then command is not supported

Following figure **illustrate checking support of few AT commands** when a Sony Ericsson mobile is connected. These commands are used in program to send the SMS.

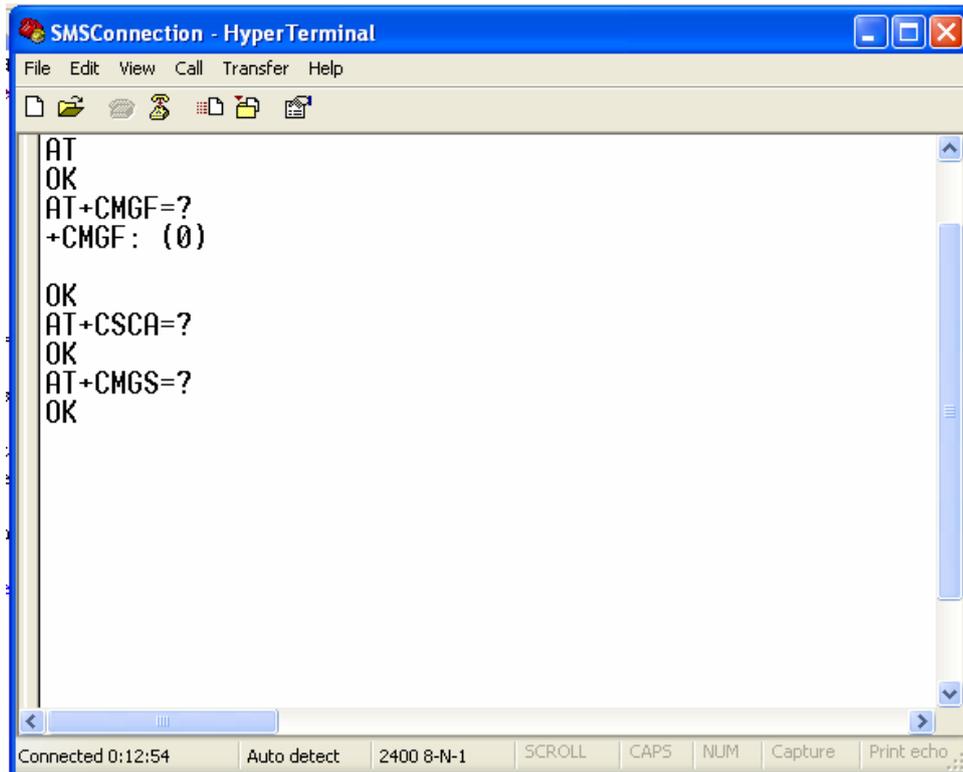


Figure 3-6 Checking commands support for connected mobile

3.1.3.4. Finding the SMSC Address

The program responsible to send SMS needs to know the SMSC address. It is possible to get the SMSC address dynamically and set it in the program variable. But as we are using two mobiles one for sending SMS and other for receiving, program does not have enough intelligence to decide, which mobile to use for which purpose. The SMSC address is required only from the handset which is used for sending SMS. Receiving is just like reading the storage device (reading unread messages from Inbox of mobile and then delete the message) and has nothing to do with SMSC.

Type the AT+CSCA command which we will use for sending SMS. The response from the mobile device will be the SMSC address, as shown below. Hardcode this received address value in the program.

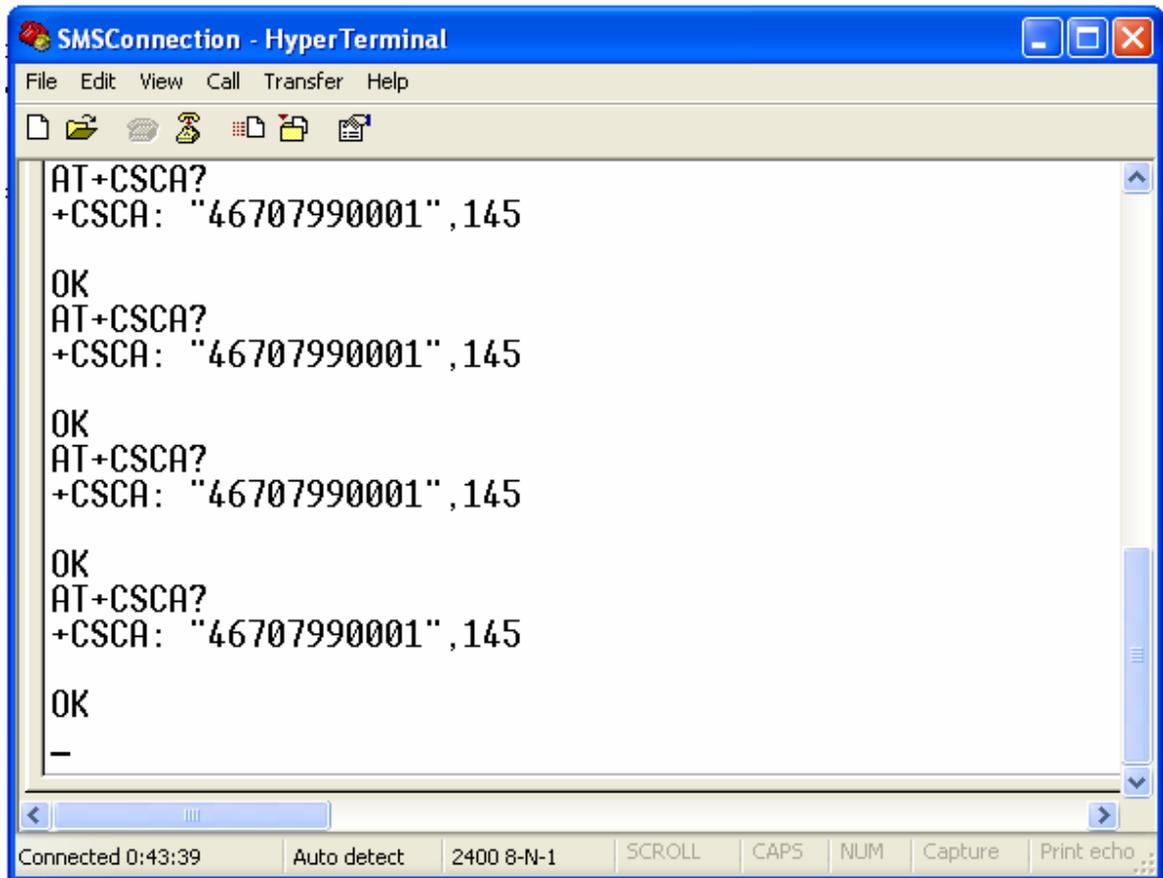


Figure 3-7 Collecting the SMSC Address

3.2 Installing the email communication sub layer

3.2.1 Email Server Installation and Configuration

- Download the trial version of winmail v4.6 as winmail.exe from <http://www.vista-files.org/programs/amax-information-technologies-inc/winmail-mail-server.html>
- Install the exe and during installation it will ask for password for administrator user 'admin' and password for user 'postmaster' which will be used later to log-in to administrator console for domain configuration, port activation and for all other administrative activities
- When the installation finishes, it asks to **Launch services** and once you select that option, it asks for mailbox address.
Put: response@dps.mdh.se where response is the username and dps.mdh.se is the domain
Then specify **password** and **confirm password** parameters with your password information and click **setup**.
It will verify and activate all required ports like SMTP(25), POP3(110),IMAP(143) and so on
- Once the above steps are executed successfully, there will be an active icon at right side at task bar. Now, double click on that and administrative console will open.
- Use the same password that you used during installation and the console will open.
- In left pane, select *User and group* → *user*, click on add button and go on creating the board member's email account as required.
- Once created, all the email accounts will be activated and can be used for sending, receiving emails.

3.2.2 Email Client Software Installation

We used Eudora v7.1 as email thick client for downloading emails from server.

- Download Eudora_7.1.0.9.exe from <http://www.eudora.com/download/>.
- Run the exe and install as normal installation
- Once the installation is over, it will ask for a short-cut at desktop. Perform the same.

- While starting for first-time, it will ask for the email server IP, username, domain and password for initial configuration and set-up.
- It will then connect to email server and your thick client configuration is perfect.

3.3 Installing the middleware

3.3.1 Installing Jboss ESB server for windows

We used JBOSS ESB server for quick learning, development, deployment and testing, instead of using JBOSS application server.

This open source JBOSS ESB package downloads are available at <http://www.jboss.org/download/>

We have used *jbossesb-server-4.4.GA* and download is available as .zip file.

The zip file needs to be unzipped in any directory (example: C:\JBoss\jbossesb-server-4.4.GA).

- We have used JDK5
- Verify JAVA_HOME points to required JDK folder appropriately

To build files and deploy, we need apache ant software and the same can be downloaded from <http://www.apache.com/> website.

We have used apache-ant-1.7.1 and the same is available as .zip file in the apache website.

- Unzip the apache ant package and specify the environment variable ANT_HOME with the ant directory path

For each JDBC adapter Listener, we need to configure the jboss-esb.xml with the following section:

```
url="jdbc:mysql://IP:port/database"
username= <USERNAME TO CONNECT TO DATABASE>
password=<PASSWORD CONNECT TO DATABASE>
tablename=<TABLE NAME THAT LISTENER WILL POLL>
status-column=<COLUMN THAT WILL CONTAIN HARDCODED VALUE 'P'; MEANS READY
TO BE POLLED BY LISTENER>
message-id-column=<UNIQUE KEY VALUE COLUMN WILL BE USED BY LISTENER DURING
DELETING THE EVENT FROM EVENT TABLE>
message-column=<COLUMN THAT WILL CONTAIN BUSINESS MESSAGE>
```

```
<sql-provider name="SQLprovider"
    url="jdbc:mysql://127.0.0.1:3306/jbpmdb"
    driver="com.mysql.jdbc.Driver"
    username="middleware"
    password="middleware">
  <sql-bus busid="helloSQLChannel" >
    <sql-message-filter
      tablename="eai_event_table"
      status-column="status"
      message-id-column="row_id"
      message-column="EMAIL_msg"
    />
  </sql-bus>
</sql-provider>
```

For EMAIL adapter Listener, we need to configure the jboss-esb.xml with the following section:

```
<property name="userName" value="response"/>
Specify the exact username of the email account that you want to poll. Here it is 'response'.
<property name="password" value="response"/>
Specify the password of the email account that you want to poll. Here it is 'response'.
<property name="pollingInterval" value="100"/>
Specify the interval period before the listener thread goes for next poll. Here it is '100'.
<property name="mailFolder" value="inbox"/>
```

Specify the email folder that needs to be polled. Here it is 'inbox'.

```
<property name="mailServer" value="127.0.0.1"/>
```

Email server IP address. Here it is 127.0.0.1

```
<activation-config>
    <property name="mailServer" value="127.0.0.1"/>
    <property name="storeProtocol" value="imap"/>
    <property name="mailFolder" value="inbox"/>
    <property name="userName" value="response"/>
    <property name="password" value="response"/>
    <property name="pollingInterval" value="100"/>
</activation-config>
```

3.3.2 Compiling and Deployment to ESB server

Go to command prompt and to the directory of shipped package of DPS.

At command prompt (at the shipped package directory) use the following command.

C:\xyz>ant deploy

This will build all the source files.

Next use

C:\xyz>ant runtest

This will deploy at ESB server

3.4 Installing the web layer

3.4.1 Installing Apache Tomcat

In order to use web part of the application, apache tomcat should be installed on the system. The apache version which we used is v5.5. Before you using tomcat, at least JRE should be installed, but JDK is recommended to be used. For installing this software, JDK should be installed on the system. This software can be downloaded free from address <http://tomcat.apache.org/>. In installing it all default setting can be used, setting a password for it is recommended and its default port is 8080 which can be changed while installing or later, if this port is used for another applications. More detail information about this software is available in this address <http://tomcat.apache.org/tomcat-5.5-doc/introduction.html>.

3.4.2 Installing MySQL

MySQL v5.0 is used for DPS database and should be installed on the system. MySQL can be downloaded from mysql site in <http://dev.mysql.com/downloads/>. Then it should be installed. In MySQL's installation, its default setting can be used. Also MySQL Connector/J — for connecting to MySQL from Java, is a jar file which is used in DPS.

After installing MySQL, database of the DPS should be created. In order to create database and filling it with initial values, a script file (DPS_db_scripts.sql) is available in DPS package, which is used for this part. The steps which should be doing in this part are as follows:

1) Testing connecting to MySQL

- In Command prompt, change directory to the MySQL path.

```
> cd ...\\MySQL_PATH\\MySQL\\MySQL Server 5.0\\bin
```

- Test connecting to database by running the following command.

```
> mysql -uroot -p
```

```
Enter password: *****
```

```
Welcome to the MySQL monitor. Commands end with ; or \g.
```

```
Your MySQL connection id is 25 to server version: 5.0.24a-community-nt
```

```
Type 'help;' or '\h' for help. Type '\c' to clear the buffer.
```

```
mysql>
```

After running 'mysql -uroot -p' command, password of the root user of mysql should be entered. If mysql is installed correctly and the password of its root user, is entered correct. It will be connected and the following messages will be shown. To exit from mysql, just 'exit' command should be runned.

If it connects to mysql correctly, next steps can be continued. Otherwise the error should be checked and solved before going to the next parts.

- 2) Creating database, related triggers, procedures and setting initial values
- Running commands to create database (sql commands in DPS_db_scripts.sql file), by running the following command:

```
> mysql -uroot -p < ... \PATH_OF_SCRIPT_FILE\DPS_db_scripts.sql"
Enter password: *****
>
```

After finishing correctly of running this command, the database and its all necessary parts will be created.

3.4.3 Run web portion of DPS

In order to run web part of DPS, the war file of web part, dps.war, is available in DPS package. In running web part of DPS, it is just needed that dps.war file is copied in the webapps folder of apache tomcat installed path, e.g. C:\Program Files\Apache Software Foundation\Tomcat 5.5\webapps. Then the apache server should be started/restarted. In order to do it, if tomcat's service is added to the current system, it is just needed to be executed form "Control Panel > Administrator Panel > Services". Otherwise it should be started from bin folder of tomcat. Now DPS web part is ready to be used by browsing this URL in a desired browser. http://IPADDRESS_OF_SYSTEM:TOMCATPORT/dps (e.g. <http://localhost:8080/dps>)