

# ePostRx Quick Start: Download, Configure, Build, Run

---

Download all necessary components, setup the system, compile, configure application server and start ePostRx.

Hardware required:

Windows 2000+, Windows XP, Windows NT with at least 1 GB memory and 40GB disk space

## 1) Download supporting applications & modules:

- a. JBOSS w/ Tomcat (currently 4.0.1) place on [DRIVE]\jboss401
  - i. e.g. **c:\jboss401**
  - ii. download from <http://anshealth.com/osdownload.htm>
  
- b. Tomcat (currently 5.0) place on [DRIVE]\tomcat50
  - i. e.g. **c:\tomcat50**
  - ii. download from <http://anshealth.com/osdownload.htm>
  
- c. SwissSQL place on [DRIVE]\swissql
  - i. e.g. **c:\swissql**
  - ii. <http://www.swissql.com/products/sqlone-apijava/sqlone-apijava-downloadform.html>
  
- d. Quartz Scheduler on [DRIVE]\quartz
  - i. e.g. **c:\quartz**
  - ii. <http://www.opensymphony.com/quartz/download.action>
  
- e. JDBC Driver ( inet JDBC drivers ) on [DRIVE]\inet
  - i. e.g. **c:\inet**
  - ii. <http://www.inetsoftware.de/Download.htm>
  
- f. JSDK 1.4 on [DRIVE]\j2sdk1.4.X\_XX
  - i. e.g. **c:\j2sdk1.4.2\_11**
  - ii. <http://java.sun.com/j2ee/1.4/download.html#sdk>
  
- g. MS SQL Server 2000+ [SQL Server 2005 has also been used]

## 2) Download the ePostRX source code:

- a. ePostRx Code unzip & place on [DRIVE]/advancenet
  - i. e.g. **c:\advancenet**
  - ii. <http://www.anshealth.com/osdownload.htm>

## 3) Orientation of ePostRX source tree:

➤ root - AdvanceNet

- com - common helper files, startup servlets and JSP taglibs
- dtts - EJB dtts

- lib - contains all external jars that support ePostRx.
- projects -
  - ansHealthB2B - contains all the B2B logic
  - ansHealthConsole - contains all the web UI logic.
  - ansHealthLib - common web shared files
  - ansHealthUtil - common Utility helper files
  - ansHealthToolkit - contains all the toolkit files
  - ansHealthReports - files that process web Reports
  - ansHealthRDBMS - contains the schema, generated Java Views, buildload scripts etc..

#### 4) Create Required DB Requirements

- Create an ePostRx DB instance & load Schema  
 The ePostRx DB Schema can be loaded manually via the MSSQL SQL Edit window by copying and applying the schema to a created database called ePostRx.
  - Load SQL from SQL file:  
 [advancenet]\projects\ansHealthRDBMS\database\mssql\schema\ansschema.txt
  - Load / create stored procedures using provided files.  
 [advancenet]\projects\ansHealthRDBMS\database\mssql\storeprocs\
- Create QUARTZ instance & load Schema
  - Load schema in [quartz]\docs\dbTables\tables\_sqlServer.sql
  - Change the DB name in the sql script to QUARTZ

#### 5) Create deployment directories

- Create [drive]\anshome & place config files here
  - e.g. c:\anshome
  - download config files from ANSHealth
    - <http://www.anshealth.com/osdownload.htm> (ANSHOME)
  - Edit ePostRxB2B\_advancenet.xml (configures the B2B module) & Edit ansHealthConsole\_advancenet.xml (configures the client/console)
    - you may need to edit :

```
<category id="LOGGING">
  <property name="LOG.PATH" value="c:/anslogs/" />
  <property name="EVENT.LEVEL" value="6" />
</category>
```

```
<category id="EPOSTRXDB">
  ...
  <property name="DB.USER" value="USERNAME" />
  <property name="DB.PASSWORD" value="PWD" />
  <property name="DB.DRIVERCLASS" value="CLASSNAME">
  <property name="DB.DRIVEURL" value="JDBC_URL">
  ...
</category>
```

```

<category id="DRUGDB">
...
<property name="DB.USER" value="USERNAME" />
<property name="DB.PASSWORD" value="PWD" />
...
<property name="DB.DRIVERCLASS" value="CLASSNAME">
<property name="DB.DRIVEURL" value="JDBC_URL">
...
</category>
<category id="SCHEDULERDB">
...
<property name="DB.USER" value="USERNAME" />
<property name="DB.PASSWORD" value="PWD" />
...
<property name="DB.DRIVERCLASS" value="CLASSNAME">
<property name="DB.DRIVEURL" value="JDBC_URL">
...
</category>

```

## 6) Create a log folder [drive]\anslogs

## 7) Set up the Application Servers

- a. **Create/set a environment variable:** ANSHOME = drive:\anshome.
- b. **JBOSS:**
  - i. Copy ans.bat & run.bat to the JBOSS/bin directory
    1. Edit ans.bat to update the location of JAVA\_HOME
    2. Edit run.bat to update JDBC\_HOME to point to your
- c. Add required jars to [DRIVE]\jboss401\server\default\lib\
  - i. Copy the following Jars into the JBoss \Default\lib directory to support ePostRx:
    1. Quartz - free scheduler (downloaded to [drive]/quartz)
    2. itext-1.3.6.jar - PDF generator (provided in src) [DRIVE]advancenet\lib\iText\
    3. SwissSqlAPI (downloaded to [drive]/swisssql/)
    4. Zql.jar - SQL parser (provided in src) [DRIVE]advancenet\lib
    5. xalan.jar - XSL transformer (provided in src) [DRIVE]advancenet\lib
    6. Xerces - advanced XML parser (provided in src) [DRIVE]advancenet\lib
    7. Apache commons jars (careful here, this may cause some conflicts)
    8. Jasper Reports - report engine (provided in src) [DRIVE]advancenet\lib\jasper
    9. Bouncy Castle JCE - crypto stuff (provided in src) [DRIVE]advancenet\lib\jce\*
    10. Cybersource Payment (ics) - credit card processor (provided in src) [DRIVE]advancenet\lib\cybersource\
- d. Create ans directory under [drive] \jboss401\server\default\deploy\ans\
- e. Modify the jbossmiminal.xml file to include the 'deploy/ans/' as a new URL deployment scanning.
  - i. [drive] \jboss401\server\default\conf\jbossmiminal.xml

- ii. Near the bottom you will see (add **red** entry):
 

```
<attribute name="URLs">
  deploy/,deploy/ans
</attribute>
```
- f. Modify the following files to change Http port from 8080 to 80.
  - i. [drive]\jboss401\server\default\deploy\jbossweb-tomcat50.sar\server.xml
  - ii. [drive]\jboss401\server\default\deploy\http-invoker.sar\META-INF\jboss-service.xml
- g. Because ePostRx uses Message Driven Beans(MDB), you will need to update the *jbossmq-destinations-service.xml* file in your configuration directory. Add the following entries for the following queues used by ePostRx (see the appendix MDB)
- h. You will need to add security to the JBoss Management Console otherwise anyone will be able to administer JBoss. Remove the JMX-Console web application from the configuration directory. In the Management /WebINF directory, update *the user.properties* file to add a new password. Also edit the web.xml and *jboss-web.xml* files to enable web application security.
- i. **(OPTIONAL)** You will need to add security to the JBoss Management Console otherwise anyone will be able to administer JBoss. Remove the JMX-Console web application from the configuration directory. In the Management /WebINF directory, update *the user.properties* file to add a new password. Also edit the web.xml and *jboss-web.xml* files to enable web application security.
- j.

## 8) Build the source

- a. Goto [drive]\advancenet\projects\ansHealthB2B\b2b\build\
  - i. modify the build.bat
    - 1. Specifically these lines
 

```
set DRIVE=c:
set ANS_HOME=%DRIVE%\advancenet
set JAVA_HOME=%DRIVE%\j2sdk1.4.0_01
```
    - 2. type: **build jboss\_deploy**  
(builds production WEB WAR - ansHealthConsole.war(precompiled JSPs for JBoss deployment ~16MB)
      - a. or **build ui\_developer**  
Builds developer/test WEB WAR - ansHealthConsole.war (with non-precompiled JSPs ~10MB). This target is mainly used for development and testing of new UI functionality. Compiles **without** the EJB interface. Can deploy to Tomcat 5.0 for testing.
- b. Goto [drive]\advancenet\projects\ansHealthB2B\b2b\build\
  - i. Modify the build.bat
    - 1. Specifically these lines
 

```
set DRIVE =c:
set ANS_HOME=%DRIVE%\advancenet

set JBOSS_HOME=%DRIVE%\jboss401
set JDBC_HOME=%DRIVE%\inet
```

set JAVA\_HOME=%DRIVE%\j2sdk1.4.2\_11

2. type: **build jboss**  
(builds JBoss EAR ansHealthB2B.ear(EAR contains the EJBs and the B2B WAR files)

## 9) Start Application Server (JBOSS)

- a. Goto [drive]\jboss401\bin
- b. Execute (in CMD window) **ans**
- c. If everything is ok, you should see an ePostRx banner that states - "\*\*\*\*EPOSTRX Go - Console started successfully\*\*\*\*".

- d. In browser goto:

<http://server:port/ansHealthConsole/htmljsp/system/login.jsp>

**user name : ansdemo**

**password : ansdemo**

**location pin: ansdemo**

## Appendix: MDB

```
<!-- ANSHEALTH QUEUES -->
```

```
<mbean code="org.jboss.mq.server.jmx.Queue"
  name="jboss.mq.destination:service=Queue,name=autocontact">
  <depends optional-attribute-
name="DestinationManager">jboss.mq:service=DestinationManager</depends>
</mbean>
```

```
<mbean code="org.jboss.mq.server.jmx.Queue"
  name="jboss.mq.destination:service=Queue,name=p2pforward">
  <depends optional-attribute-
name="DestinationManager">jboss.mq:service=DestinationManager</depends>
</mbean>
```

```
<mbean code="org.jboss.mq.server.jmx.Queue"
  name="jboss.mq.destination:service=Queue,name=inventoryverification">
  <depends optional-attribute-
name="DestinationManager">jboss.mq:service=DestinationManager</depends>
</mbean>
```

```
<mbean code="org.jboss.mq.server.jmx.Queue"
  name="jboss.mq.destination:service=Queue,name=scheduler">
  <depends optional-attribute-
name="DestinationManager">jboss.mq:service=DestinationManager</depends>
</mbean>
```

```
<mbean code="org.jboss.mq.server.jmx.Queue"
  name="jboss.mq.destination:service=Queue,name=schedulerjob">
  <depends optional-attribute-
name="DestinationManager">jboss.mq:service=DestinationManager</depends>
</mbean>
```

```
<mbean code="org.jboss.mq.server.jmx.Queue"
  name="jboss.mq.destination:service=Queue,name=rule">
  <depends optional-attribute-
name="DestinationManager">jboss.mq:service=DestinationManager</depends>
</mbean>
```

```
<mbean code="org.jboss.mq.server.jmx.Queue"
  name="jboss.mq.destination:service=Queue,name=resourceconnector">
  <depends optional-attribute-
name="DestinationManager">jboss.mq:service=DestinationManager</depends>
</mbean>
```

```
<mbean code="org.jboss.mq.server.jmx.Queue"
  name="jboss.mq.destination:service=Queue,name=orderrx">
  <depends optional-attribute-
name="DestinationManager">jboss.mq:service=DestinationManager</depends>
</mbean>
```

```
<mbean code="org.jboss.mq.server.jmx.Queue"
  name="jboss.mq.destination:service=Queue,name=retailrx">
  <depends optional-attribute-
name="DestinationManager">jboss.mq:service=DestinationManager</depends>
```

</mbean>

<mbean code="org.jboss.mq.server.jmx.Queue"

name="jboss.mq.destination:service=Queue,name=escript">

<depends optional-attribute-name="DestinationManager">jboss.mq:service=DestinationManager</depends>

</mbean>

<mbean code="org.jboss.mq.server.jmx.Queue"

name="jboss.mq.destination:service=Queue,name=inventorysupplierpo">

<depends optional-attribute-name="DestinationManager">jboss.mq:service=DestinationManager</depends>

</mbean>

<mbean code="org.jboss.mq.server.jmx.Queue"

name="jboss.mq.destination:service=Queue,name=utilities">

<depends optional-attribute-name="DestinationManager">jboss.mq:service=DestinationManager</depends>

</mbean>

<mbean code="org.jboss.mq.server.jmx.Queue"

name="jboss.mq.destination:service=Queue,name=ivr">

<depends optional-attribute-name="DestinationManager">jboss.mq:service=DestinationManager</depends>

</mbean>

<mbean code="org.jboss.mq.server.jmx.Queue"



```
name="jboss.mq.destination:service=Queue,name=rhubbulkload">
  <depends optional-attribute-
name="DestinationManager">jboss.mq:service=DestinationManager</depends>
</mbean>
```

```
<mbean code="org.jboss.mq.server.jmx.Queue"
  name="jboss.mq.destination:service=Queue,name=ssbulkload">
  <depends optional-attribute-
name="DestinationManager">jboss.mq:service=DestinationManager</depends>
</mbean>
```

8. There is an application wide JMS topic that needs to be set up as well. This is used to allow all ePostRx applications(B2B) to talk to each other. You will need to update the *jbossmq-destinations-service.xml* file in your configuration directory. Add the following entries for the following topics used by ePostRx:

```
<!-- ANS TOPICS-->
<mbean code="org.jboss.mq.server.jmx.Topic"
  name="jboss.mq.destination:service=Topic,name=epostrx">
  <depends optional-attribute-
name="DestinationManager">jboss.mq:service=DestinationManager</depends>
</mbean>
```