

openCRX Installation Guide for JBoss 4 and JBoss 5

Version 2.3



www.opencrx.org

License

The contents of this file are subject to a BSD license (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at <http://www.opencrx.org/license.htm>

Copyright 2008 © CRIXP Corp. All rights reserved.



Table of Contents

1	About this Book.....	3
1.1	Who this book is for.....	3
1.2	What do you need to understand this book.....	3
1.3	Tips, Warnings, etc.....	3
2	Prerequisites.....	4
2.1	JDK 5.0.....	4
2.2	JBoss.....	5
2.3	openCRX SDK.....	5
2.4	Database.....	5
3	Installing openCRX for JBoss.....	6
4	Configuring Container Security.....	9
4.1	Configuring LoginModule.....	9
5	Starting JBoss.....	11
6	Additional JBoss Settings.....	12
6.1	Port Binding and Security.....	12
7	Next Steps.....	13

List of Figures

List of Listings

Listing 1:	The file server.log.properties.....	7
Listing 2:	The file run.bat on Windows.....	7
Listing 3:	The file run.sh on Linux.....	8
Listing 4:	The file run.conf on Linux.....	8
Listing 5:	Connector definition in server.xml.....	8
Listing 6:	JBoss configuration for JAAS based authentication.....	10
Listing 7:	File users.properties with syntax user=password.....	10
Listing 8:	File roles.properties with syntax user.Roles=role1,role2.....	10
Listing 9:	JBoss console output.....	11

1 About this Book

openCRX is the leading open source CRM enterprise suite. openCRX is based on the openMDX application framework, an open source application framework based on the OMG's model driven architecture (MDA) standards, delivering maximum openness, standards compliance and a state-of-the-art component-based architecture.

This book describes the installation of openCRX for the JBoss application server.

1.1 Who this book is for

The intended audience are openCRX administrators and system administrators who either want to deploy openCRX on the JBoss application server.

1.2 What do you need to understand this book

This book describes the installation of openCRX for JBoss. The book assumes that you are familiar with JBoss deployment concepts and administration.

1.3 Tips, Warnings, etc.

We make use the following pictograms:



Information provided as a "Tip" might be helpful for various reasons: time savings, risk reduction, etc.



You should carefully read information marked with "Important". Ignoring such information is typically not a good idea.



Warnings should not be ignored (risk of data loss, etc.)

2 Prerequisites

As a first step select the openCRX version you want to install. Based on the published version compatibility information you can determine the appropriate versions of openMDX, JBoss, and Java JDK:

<http://www.opencrx.org/faq.htm#versioncompatibility>



Tip

Write down the version numbers of the software packages you have chosen to install – this may be helpful in the future in case you require support or want to file a bug report:

openCRX v _____
openMDX v _____
JBoss v _____
JDK v _____

2.1 JDK 5.0

Install the Sun Java JDK 5.0, available from the following site:

http://java.sun.com/javase/downloads/index_jdk5.jsp



Warning

If you deploy openCRX on Sun's JVM you might run into a problem that is mind-boggling (but specific to the Sun JVM). It's the dreaded

PermGen OutOfMemoryError

Lots of people have been struggling with this error (which is by the way not related to openCRX – any application could trigger it) and as of today we have not come across a solution other than to avoid Sun's JVM (JVMs by Oracle/BEA and IBM are – by construction – not prone to this error). Do a Google search if you want to convince yourself or drop us a line if you have a solution. But be aware that increasing the memory allocated for the permanent generation is a bogus solution – it will only postpone the problem:

- <http://my.opera.com/karmazilla/blog/2007/03/13/good-riddance-permgen-outofmemoryerror>
- <http://my.opera.com/karmazilla/blog/2007/03/15/permgen-strikes-back>
- <http://crashingdaily.wordpress.com/2007/02/04/crashing-tomcat/>



Important

It is not sufficient to have a Java Runtime Environment (JRE) only. The full-blown JDK is required to run openCRX.



Tip

Don't forget to set the **environment variable** `JAVA_HOME`. It should point to the installation directory of your JDK, e.g. `D:\Java\jdk1.5.0` on Windows or `/opt/Java/jdk1.5.0` on Linux.

2.2 JBoss

Download **JBoss 4 or JBoss 5** from <http://www.jboss.org/jbossas/downloads/>



Do not download the JBoss version that includes JDK 6 (e.g. jboss-5.x.x.GA-jdk6.zip) as openCRX requires JDK 5. Get the plain-vanilla distribution jboss-5.x.x.GA.zip!



As there are a few known issues with JBoss 5, use JBoss 4 (e.g. JBoss 4.2.2.GA) if you don't want to deal with the issues of JBoss 5.

For more information, see <http://www.opencrx.org/faq.htm#deploymentmatrix>

2.3 openCRX SDK

Download and install the **openCRX SDK**: <http://www.opencrx.org/sdk.htm>
Installation instructions are also available from the above website.



If your platform is not supported by the openCRX SDK Installer, you can install the openCRX SDK on one of the supported platforms and then copy the required files to your target platform.

2.4 Database

Please note that you **must set up the openCRX database** as described in the respective openCRX database installation guide **before you continue**. For example, if you want to install openCRX for PostgreSQL you must first install PostgreSQL and the matching openCRX database definitions.

Database installation guides for the supported database management systems are available from <http://www.opencrx.org/documents.htm>

You will also need **datasource configuration files**. Sample data source configurations for various database management systems are contained in the openCRX SDK in the directory

`<SDK_Install_Dir>\opencrx-2.3.0\core\src\connector\openmdx-2`



Please remember that you will also need a **JDBC driver** appropriate for your database management system (please refer to the relevant DB installation guide for additional information).

Once you have successfully installed the database you are ready to continue with the JBoss setup.

3 Installing openCRX for JBoss

In a first step, install JBoss by extracting the JBoss distribution a directory of your choice, e.g. `D:\jboss` on Windows or `/opt/jboss` on Linux. From now on we will refer to this directory as `JBOSS_HOME`.



On Windows, avoid paths that contain blanks (e.g. the famous `C:\Program File\...`) as there are all kinds of pitfalls down the road...



Make sure that you add `JAVA_HOME` to your system environment variables as JBoss needs the JDK to compile JSPs, e.g. `JAVA_HOME=D:\java\jdk1.5.0` on Windows or `JAVA_HOME=/opt/java/jdk1.5.0` on Linux.

Next you must deploy openCRX to JBoss. You do this by copying several files to the JBoss server directory structure:

- Copy the file `openmdx-kernel.jar` from the directory `<SDK_Install_Dir>\opencrx-2.3.0\opt\openmdx-2.3.0\jre-1.5\core\lib` to the JBoss directory `JBOSS_HOME\server\default\lib`.
- Copy the appropriate database **JDBC driver** to the JBoss directory `JBOSS_HOME\server\default\lib`. The openCRX database installation manual describes how to download the drivers.
- Copy the file `opencrx-core-CRX.ear` from the directory `<SDK_Install_Dir>\opencrx-2.3.0\jre-1.5\core\deployment-unit` to the JBoss directory `JBOSS_HOME\server\default\deploy`.



Do NOT deploy any other openCRX EARs (e.g. groupware) before having completed the openCRX QuickStart guide.

Optionally, you can expand the file `opencrx-core-CRX.ear` to the directory `JBOSS_HOME\server\default\deploy\opencrx-core-CRX.ear`

Optionally, you can also expand the file `opencrx-core-CRX.war` to the directory `JBOSS_HOME\server\default\deploy\opencrx-core-CRX.ear\opencrx-core-CRX.war`

- Install the **datasource configuration file** appropriate for your DBMS, e.g. `jdbc-opencrx-CRX-postgresql-ds.xml` for PostgreSQL, to the directory `JBOSS_HOME\server\default\deploy`. Sample datasource configuration files are available in the directory `<SDK_Install_Dir>\opencrx-2.3.0\core\src\connector\jboss-3`



Verify all the relevant information for correctness, e.g. ensure that the connection url – in particular the database name – the user name, and the password match with your installation. A simple copy/paste of our sample files will typically not work, i.e. it is expected that you will have to adapt some of the parameters.

- Copy the file **slf4j-openmdx1.jar** from the openCRX SDK installation directory `<SDK_Install_Dir>\opencrx-2.3.0\opt\openmdx-2.3.0\jre-1.5\log\lib` to the JBOSS directory `JBOSS_HOME\server\default\lib`.
- create the file **server.log.properties** with the following content in the JBOSS directory `JBOSS_HOME\server\default`:

Listing 1: The file server.log.properties

```
ApplicationId = openCRX
LogFileExtension = log
LogFilePath = D:/jboss/server/default/log/
LogLevel = warning
java.LoggingMechanism = SharedDatedFileLoggingMechanism
LogFormat = ${logger}|${time}|${level}|${logsource}|${host}|${thread}|${class}|${method}|${line}|${summary}|${detail}
```



Adapt `D:/jboss/server/default` to your environment (on Linux this might be `/opt/jboss/server/default`).

Notice the slashes ("`/`") in the path – they are forward slashes on all platforms!

Next you must edit the file `JBOSS_HOME\bin\run.bat`. We will add a few commands to clean various directories before starting JBoss to ensure a clean startup environment. Furthermore, we also set a few Java VM options which are required for the openMDX application framework.

On Windows add the following lines to the beginning of the file `JBOSS_HOME\bin\run.bat` and then look for the comment

```
rem Setup JBoss specific properties
```

to change/add various Java VM options as shown below:

Listing 2: The file run.bat on Windows

```
RMDIR /S /Q D:\jboss\server\default\tmp
RMDIR /S /Q D:\jboss\server\default\work
@echo off
rem -----
rem JBoss Bootstrap Script for Windows
rem -----
...
rem Setup JBoss specific properties
rem JVM memory allocation pool parameters. Modify as appropriate.
set JAVA_OPTS=%JAVA_OPTS% -Xms128m -Xmx800m -XX:MaxPermSize=256m

rem Setup openMDX-specific properties
set JAVA_OPTS=%JAVA_OPTS% -Djava.protocol.handler.pkgs=org.openmdx.kernel.url.protocol
set JAVA_OPTS=%JAVA_OPTS% -Dorg.openmdx.log.config.filename=D:\jboss\server\default\server.log.properties
set JAVA_OPTS=%JAVA_OPTS% -Dorg.opencrx.maildir=D:\jboss\server\default\maildir
...
```



Adapt `D:\jboss\server\default` to your environment and make sure that there are no line breaks in the set commands. Each `-D` options is of the form `-Dname=value` and must be on a single line.

On Linux add the following lines (to remove the directories tmp and work) to the beginning of the file `JBOSS_HOME\bin\run.sh`:

Listing 3: The file run.sh on Linux

```
#!/bin/sh
cd /opt/jboss/server/default
rm -Rf tmp
rm -Rf work
...
```



Adapt `/opt/jboss/server/default` to your environment.

On Linux open the file `JBOSS_HOME\bin\run.conf` and then look for the comment `# Specify options to pass to the Java VM` to change/add various Java VM options as shown below:

Listing 4: The file run.conf on Linux

```
...
# Specify options to pass to the Java VM.
if [ "x$JAVA_OPTS" = "x" ]; then
    JAVA_OPTS="-Xms128m -Xmx800m -XX:MaxPermSize=256m ..."
fi

rem Setup openMDX-specific properties
JAVA_OPTS="$JAVA_OPTS -Djava.protocol.handler.pkgs=org.openmdx.kernel.url.protocol"
JAVA_OPTS="$JAVA_OPTS -Dorg.openmdx.log.config.filename=/opt/jboss/server/default/server.log.properties"
JAVA_OPTS="$JAVA_OPTS -Dorg.openmdx.maildir=/opt/jboss/server/default/maildir"

# Sample JPDA settings for remote socket debugging
JAVA_OPTS="$JAVA_OPTS -Xrunjdpw:transport=dt_socket,address=8787,server=y,suspend=n"
...
```



If you are using PostgreSQL and if you set up the database as proposed by us, you can also add the following JVM option (to `run.bat` on Windows or `run.conf` on Linux) to enhance performance:
`-Dorg.openmdx.persistence.jdbc.useLikeForOidMatching=false`

Finally, we have to make a change to JBoss's `server.xml` for full UTF-8 support:

- open the file `JBOSS_HOME\server\default\deploy\jbossweb.sar\server.xml`
- look for the connector definition `<Connector protocol="HTTP/1.1" port="8080" address="{jboss.bind.address}"` and add the option `URIEncoding="UTF-8"` so that the connector definition looks as shown below:

Listing 5: Connector definition in server.xml

```
<Connector protocol="HTTP/1.1" port="8080" address="{jboss.bind.address}"
    connectionTimeout="20000" redirectPort="8443" URIEncoding="UTF-8" />
```


4 Configuring Container Security

As a final step you must activate container security for the openCRX application. You can configure the file-based *UsersRolesLoginModule* and/or the database-based *DatabaseServerLoginModule*.

4.1 Configuring LoginModule



openCRX stores security information in the database tables **OOCSE1_***, i.e.

OOCSE1_Authenticationcontext	OOCSE1_Authenticationcontext_
OOCSE1_Credential	OOCSE1_Credential_
OOCSE1_Permission	OOCSE1_Permission_
OOCSE1_Policy	OOCSE1_Policy_
OOCSE1_Principal	OOCSE1_Principal_
OOCSE1_Privilege	OOCSE1_Privilege_
OOCSE1_Realm	OOCSE1_Realm_
OOCSE1_Role	OOCSE1_Role_
OOCSE1_Segment	OOCSE1_Segment_
OOCSE1_Subject	OOCSE1_Subject_

JBoss can be enabled to access these tables by configuring the database login module. This way users can be managed in openCRX and are immediately available as JBoss logins.



It is strongly recommended that you stay with the file-based *UsersRolesLoginModule* for the user **admin-Root**. This simplifies the openCRX bootstrapping.



We recommend that you stay with the file-based authentication for all users until you have finished installing openCRX. You avoid situations where you have to trouble-shoot multiple issues at the same time...

Activate JAAS based authentication by adding the following configuration entries for the openCRX servlet to the JBoss configuration file

JBOSS_HOME\server\default\conf\login_config.xml
(**login-config.xml** on Unix platforms)

Listing 6: JBoss configuration for JAAS based authentication

```
<application-policy name="opencrx-core-CRX">
  <authentication>
    <login-module code="org.jboss.security.auth.spi.DatabaseServerLoginModule" flag="sufficient">
      <module-option name="dsJndiName">java:/jdbc_opencrx_CRX</module-option>
      <module-option name="principalsQuery">SELECT c.passwd FROM OOCSE1_PRINCIPAL p, OOCSE1_CREDENTIAL c WHERE
(p.object_id LIKE 'principal/CRX/Root/Default/%') AND (p.credential = c.object_id) AND (p.name = ?)</module-
option>
      <module-option name="rolesQuery">SELECT r.name, 'Roles' FROM OOCSE1_PRINCIPAL_pg, OOCSE1_PRINCIPAL p,
OOCSE1_PRINCIPAL_pn, OOCSE1_ROLE r WHERE (p.object_id = pn.object_id) AND (pn.is_member_of = pg.object_id) AND
(pg.granted_role = r.object_id) AND (p.object_id LIKE 'principal/CRX/Root/Default/%') AND (p.name = ?)</module-
option>
      <module-option name="ignorePasswordCase">>true</module-option>
      <module-option name="hashCharset">UTF-8</module-option>
      <module-option name="hashEncoding">base64</module-option>
      <module-option name="hashAlgorithm">MD5</module-option>
    </login-module>
    <login-module code="org.jboss.security.auth.spi.UsersRolesLoginModule" flag="sufficient" >
      <module-option name="usersProperties">users.properties</module-option>
      <module-option name="rolesProperties">roles.properties</module-option>
    </login-module>
  </authentication>
</application-policy>
```

The settings above enable both **file-based authentication** and **database-based authentication**.

Next you need to create the files **users.properties** and **roles.properties** in the directory **JBOSS_HOME\server\default\conf**.

*Listing 7: File **users.properties** with syntax **user=password***

```
admin-Root=rootSecret
admin-Standard=adminSecret
guest=guest
```

*Listing 8: File **roles.properties** with syntax **user.Roles=role1,role2***

```
admin-Root.Roles=OpenCrxRoot
admin-Standard.Roles=OpenCrxAdministrator
guest.Roles=OpenCrxUser
```



It is strongly recommended that you stay with the file-based UsersRolesLoginModule for the user **admin-Root**. This simplifies the openCRX bootstrapping.

5 Starting JBoss

You are now ready to start JBoss. Open a command shell and start **JBOSS_HOME\bin\run.bat**. You should verify that the start options match the ones described earlier:

Listing 9: JBoss console output

```
[ 9:32:02.91]D:\jboss\bin>run
=====
JBoss Bootstrap Environment

JBOSS_HOME: D:\jboss

JAVA: D:\Java\jrockit1.5.0\bin\java

JAVA_OPTS:
-Dprogam.name=run.bat
-Xms128m
-Xmx800m
-XX:MaxPermSize=256m
-Djava.protocol.handler.pkgs=org.openmdx.kernel.url.protocol
-Dorg.openmdx.log.config.filename=D:\jboss\server\default\server.log.properties
-Dorg.openmdx.maildir=D:\jboss\server\default\maildir
-Dorg.openmdx.persistence.jdbc.useLikeForOidMatching=false
-Dorg.jboss.resolver.warning=true
-Dsun.rmi.dgc.client.gcInterval=3600000
-Dsun.rmi.dgc.server.gcInterval=3600000

CLASSPATH: D:\jboss\bin\run.jar
=====

[WARN ] -XX:MaxPermSize=256m is not a valid VM option. Ignoring
09:32:10,193 INFO [ServerImpl] Starting JBoss (Microcontainer)...
09:32:10,209 INFO [ServerImpl] Release ID: JBoss [Morpheus] 5.0.0.GA (build: SVNTag=JBoss_5_0_0_GA date=200812041714)
09:32:10,209 INFO [ServerImpl] Bootstrap URL: null
09:32:10,209 INFO [ServerImpl] Home Dir: D:\jboss
09:32:10,209 INFO [ServerImpl] Home URL: file:/D:/jboss/
09:32:10,209 INFO [ServerImpl] Library URL: file:/D:/jboss/lib/
09:32:10,224 INFO [ServerImpl] Patch URL: null
09:32:10,224 INFO [ServerImpl] Common Base URL: file:/D:/jboss/common/
09:32:10,224 INFO [ServerImpl] Common Library URL: file:/D:/jboss/common/lib/
09:32:10,224 INFO [ServerImpl] Server Name: default
09:32:10,224 INFO [ServerImpl] Server Base Dir: D:\jboss\server
09:32:10,224 INFO [ServerImpl] Server Base URL: file:/D:/jboss/server/
09:32:10,224 INFO [ServerImpl] Server Config URL: file:/D:/jboss/server/default/conf/
09:32:10,224 INFO [ServerImpl] Server Home Dir: D:\jboss\server\default
...
...
09:33:15,348 INFO [BaseLocalProxyFactory] Bound EJB LocalHome 'opencrx_core_CRX_Gateway' to jndi 'local/opencrx_core_CRX_Gateway@3067796'
09:33:15,364 INFO [ProxyFactory] Bound EJB Home 'opencrx_core_CRX_Gateway' to jndi 'org.opencrx.core.CRX.Gateway'
09:33:15,380 INFO [EjbModule] Deploying opencrx_core_CRX_kernel_mandatory
09:33:15,395 WARN [EjbModule] EJB configured to bypass security. Please verify if this is intended. Bean=opencrx_core_CRX_kernel_mandatory
Deployment-vfszip:/D:/jboss/server/default/deploy/opencrx-core-CRX.ear/kernel.jar
09:33:15,473 INFO [BaseLocalProxyFactory] Bound EJB LocalHome 'opencrx_core_CRX_kernel_mandatory' to jndi
'org.opencrx.core.CRX.local.mandatory.kernel'
09:33:15,505 INFO [EjbModule] Deploying opencrx_core_CRX_security_mandatory
09:33:15,505 WARN [EjbModule] EJB configured to bypass security. Please verify if this is intended. Bean=opencrx_core_CRX_security_mandatory
Deployment-vfszip:/D:/jboss/server/default/deploy/opencrx-core-CRX.ear/security.jar
09:33:15,520 INFO [BaseLocalProxyFactory] Bound EJB LocalHome 'opencrx_core_CRX_security_mandatory' to jndi
'org.opencrx.core.CRX.local.mandatory.security'
09:33:15,552 INFO [Http11Protocol] Starting Coyote HTTP/1.1 on http-127.0.0.1-8080
09:33:15,583 INFO [AjpProtocol] Starting Coyote AJP/1.3 on ajp-127.0.0.1-8009
09:33:15,598 INFO [ServerImpl] JBoss (Microcontainer) [5.0.0.GA (build: SVNTag=JBoss_5_0_0_GA date=200812041714)] Started in 1m:5s:358ms
```

Now you are ready to continue with the openCRX QuickStart Guide.

6 Additional JBoss Settings

6.1 Port Binding and Security

Newer versions of JBoss (4.2 and newer) bind to localhost only by default (it used to be global 0.0.0.0). You can get this same behavior by starting JBoss with **-b 0.0.0.0**.

If you look in the readme.html in your JBoss distribution there is a link with info on how to secure JBoss once you bind to 0.0.0.0. Alternatively, you can fiddle with the Tomcat options in the file **server.xml**.

7 Next Steps

Now that you have successfully deployed openCRX on your application server you can continue with the openCRX QuickStart guide.