

# openCRX Installation Guide for JBoss 4

Version 1.7.1



[www.opencrx.org](http://www.opencrx.org)

**openCRX Installation Guide for JBoss 4: Version 1.7.1**

by [www.opencrx.org](http://www.opencrx.org)

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 *here* (<http://www.opencrx.org/license.htm>)

# Table of Contents

<b>1. About this Book</b> .....	<b>1</b>
Who this book is for .....	1
What do you need to understand this book.....	1
<b>2. Prerequisites</b> .....	<b>2</b>
<b>3. Installing openCRX for JBoss</b> .....	<b>3</b>
<b>4. Configuring Security</b> .....	<b>5</b>
Configuring UsersRolesLoginModule .....	5
Configuring DatabaseServerLoginModule.....	6
<b>5. Starting JBoss</b> .....	<b>8</b>
<b>6. Install openCRX as Windows Service</b> .....	<b>9</b>
<b>7. Install JBoss as daemon and Startup Service on Linux</b> .....	<b>10</b>
<b>A. Appendix</b> .....	<b>11</b>
<b>Bibliography</b> .....	<b>12</b>

# List of Examples

3-1. listing of server.log.properties. ....	3
3-2. Java VM options required for openMDX on Windows .....	4
3-3. Java VM options required for openMDX on Linux.....	4
4-1. JBoss login_config.xml for JAAS login configuration for the root servlet. ....	5
4-2. JBoss login_config.xml for JAAS login configuration for the standard servlet. ....	5
4-3. openCRX.users.properties with user=password syntax.....	5
4-4. openCRX.roles.properties with user.Roles=role1,role2 syntax.....	5
4-5. JBoss login_config.xml for JAAS login configuration for the root servlet .....	6
4-6. openCRX.users.properties with user=password syntax.....	6
4-7. openCRX.roles.properties with user.Roles=role1,role2 syntax.....	6
4-8. JBoss login_config.xml for JAAS login configuration for the standard servlet for NON-PostgreSQL databases.....	6
4-9. JBoss login_config.xml for JAAS login configuration for the standard servlet for PostgreSQL databases. ....	7
5-1. JBoss start options. ....	8
5-2. Deployment of database datasource. ....	8
5-3. Deployment of opencrx-core-CRX-App.ear.....	8
5-4. Deployment of opencrx-core-CRX-web.ear and opencrx-core-CRX-Root-web.ear .....	8
5-5. JBoss running – last few lines of console output.....	8
6-1. Starting JBoss as Windows service. ....	9

# Chapter 1. About this Book

*openCRX* is the leading open source CRM tool. *openCRX* is based on the *openMDX [02]* application framework, an open source application framework based on the OMG's model driven architecture (MDA) standards. This guarantees maximum openness, standards compliance and a state-of-the-art component-based architecture.

## Who this book is for

This book describes the installation of *openCRX* for *JBoss*. The intended audience are *openCRX* and application server system administrators.

## 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.

## Chapter 2. Prerequisites

As a first step select the *openCRX* version you want to install. Based on the published *version compatibility information* (<http://www.opencrx.org/faq.htm#versioncompatibility>) you can determine the appropriate versions of *openMDX*, *JBoss*, and *Java JDK/JRE*.

- Download **JBoss** from *here* (<http://www.jboss.org/downloads/index>)
- Download **openMDX** from *here* ([http://sourceforge.net/project/showfiles.php?group\\_id=75132](http://sourceforge.net/project/showfiles.php?group_id=75132)).
- Download **openCRX** from *here* ([http://sourceforge.net/project/showfiles.php?group\\_id=95219](http://sourceforge.net/project/showfiles.php?group_id=95219)). You must download the *opencrx-core* distribution (e.g. *opencrx-1.7.0-core.CRX.jre-1.4.zip*).



**Important** As a first step you must install the database as described in the database distribution. E.g. if you want to install *openCRX* for *MySQL* you must first install *MySQL* and the *openCRX* database definitions. If you have successfully installed the database you are ready to continue with the *JBoss* setup.

# Chapter 3. Installing openCRX for JBoss

In a first step you must install *JBoss* by extracting the delivered *JBoss* distribution to your program directory, e.g. `d:\pgm\jboss-4.0.1` on Windows or `/opt/jboss` on Linux or any other Posix OS.



Make sure that you add **JAVA\_HOME** to your system environment variables, e.g. `JAVA_HOME=D:\pgm\jdk1.4.2` on Windows or `JAVA_HOME=/usr/java/j2sdk1.4.2` on Linux. `JAVA_HOME` is required by *JBoss* in order to compile JSPs.

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

- Copy the file `openmdx-kernel.jar` contained in the *openMDX* distribution to the directory `d:\pgm\jboss-4.0.1\server\default\lib` on Windows or `/opt/jboss/server/default/lib` on Linux.



If you are upgrading from any version older than *openCRX* v1.3.0 you must **remove/delete** the file `openmdx-base.jar` from the directory `d:\pgm\jboss-4.0.1\server\default\lib`.

- Copy the appropriate database JDBC driver to the directory `d:\pgm\jboss-4.0.1\server\default\lib`. The *openCRX* database installation manual describes how to download the drivers (e.g. for *MySQL* the JDBC driver comes as `mysql-connector-java-3.0.16.jar`)
- Copy the file `opencrx-core-CRX-App.ear` contained in the *openCRX* distribution to the directory `d:\pgm\jboss-4.0.1\server\default\deploy` on Windows or `/opt/jboss/server/default/deploy` on Linux
- Copy the file `opencrx-core-CRX-web.ear` contained in the *openCRX* distribution to the directory `d:\pgm\jboss-4.0.1\server\default\deploy` on Windows or `/opt/jboss/server/default/deploy` on Linux. You can also open `opencrx-core-CRX-web.ear` with a ZIP utility and extract the content to the directory `d:\pgm\jboss-4.0.1\server\default\deploy\opencrx-core-CRX-web.ear` on Windows or `/opt/jboss/server/default/deploy/opencrx-core-CRX-web.ear` on Linux. If you want to edit the content of the file `opencrx-core-CRX.war` without the zip/unzip roundtrip you can also extract the content of that file with a ZIP utility.
- Copy the file `opencrx-core-CRX-Root-web.ear` contained in the *openCRX* distribution to the directory `d:\pgm\jboss-4.0.1\server\default\deploy` on Windows or `/opt/jboss/server/default/deploy` on Linux. You can also open `opencrx-core-CRX-Root-web.ear` with a ZIP utility and extract the content to the directory `d:\pgm\jboss-4.0.1\server\default\deploy\opencrx-core-CRX-Root-web.ear` on Windows or `/opt/jboss/server/default/deploy/opencrx-core-CRX-Root-web.ear` on Linux. If you want to edit the content of the file `opencrx-core-CRX-Root.war` without the zip/unzip roundtrip you can also extract the content of that file with a ZIP utility.
- Install the datasource configuration file, e.g. copy the file `jdbc-opencrx-CRX-mysql-ds.xml` (if you use *openCRX* with *MySQL*) contained in the file `opencrx-core.jboss-3-connector.zip` of the *openCRX* distribution to the directory `d:\pgm\jboss-4.0.1\server\default\deploy` on Windows or `/opt/jboss/server/default/deploy` on Linux.
- Create the file `d:\pgm\jboss-4.0.1\server\default\server.log.properties` on Windows or `/opt/jboss/server/default/server.log.properties` on Linux with the following content.

### Example 3-1. listing of server.log.properties.

```
ApplicationId = openCRX
LogFileExtension = log
LogFilePath = D:/pgm/jboss-4.0.1/server/default/log/
LogLevel = warning
java.LoggingMechanism = SharedDatedFileLoggingMechanism
```



Adapt **D:/pgm/jboss-4.0.1** to your environment!

Next you must set a few Java VM options which are required for the *openMDX* application framework.

On Windows add the following lines to *d:\pgm\jboss-4.0.1\bin\run.bat* after the lines indicated below. Also uncomment the line *set JAVA\_OPTS=%JAVA\_OPTS% -Xms128m -Xmx512m*. This gives more memory to the Java VM (and depending on your environment you may want to increase the value of the option *Xmx*).

### Example 3-2. Java VM options required for openMDX on Windows

```
rem Sun JVM memory allocation pool parameters. Uncomment and modify as appropriate.
set JAVA_OPTS=%JAVA_OPTS% -Xms128m -Xmx512m

rem Setup openMDX-specific properties
set JAVA_OPTS=%JAVA_OPTS% -Dorg.openmdx.compatibility.base.application.j2ee.domain=apps
set JAVA_OPTS=%JAVA_OPTS% -Dorg.openmdx.compatibility.base.application.j2ee.server=server1
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:\pgm\jboss-4.0.1\server\default\server.log.properties
set JAVA_OPTS=%JAVA_OPTS% -Dmail.SSLSocketFactory.class=org.opencrx.kernel.mail.SendMailSSLSocketFactory
```



**Important!** Adapt *D:\pgm\jboss-4.0.1* to your environment!



**Important!** 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 */opt/jboss/bin/run.conf* towards the end of the file.

### Example 3-3. Java VM options required for openMDX on Linux

```
# Setup openMDX-specific properties
JAVA_OPTS="$JAVA_OPTS -Xms128m -Xmx512m"
JAVA_OPTS="$JAVA_OPTS -Dorg.openmdx.compatibility.base.application.j2ee.domain=apps"
JAVA_OPTS="$JAVA_OPTS -Dorg.openmdx.compatibility.base.application.j2ee.server=server1"
JAVA_OPTS="$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 -Dmail.SSLSocketFactory.class=org.opencrx.kernel.mail.SendMailSSLSocketFactory"
```



**Important!** Adapt */opt/jboss/server/default/* to your environment!



**Important!** Make sure that there are **no line breaks inside options**. Each *-D* options is of the form *-Dname=value* and **must be on a single line**.

# Chapter 4. Configuring Security

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

## Configuring UsersRolesLoginModule

Activate JAAS based authentication by adding the following **TWO** configuration entries for the standard and the root servlet to *d:\pgm\jboss-4.0.1\server\default\conf\login\_config.xml* (*login-config.xml* on Unix platforms!).

Add the following security policy for the **root** servlet:

### Example 4-1. JBoss login\_config.xml for JAAS login configuration for the root servlet.

```
<application-policy name = "opencrx-core-CRX-Root">
  <authentication>
    <login-module code="org.jboss.security.auth.spi.UsersRolesLoginModule" flag = "required" >
      <module-option name="usersProperties">openCRX.users.properties</module-option>
      <module-option name="rolesProperties">openCRX.roles.properties</module-option>
    </login-module>
  </authentication>
</application-policy>
```

Add the following security policy for the **standard** servlet:

### Example 4-2. JBoss login\_config.xml for JAAS login configuration for the standard servlet.

```
<application-policy name = "opencrx-core-CRX">
  <authentication>
    <login-module code="org.jboss.security.auth.spi.UsersRolesLoginModule" flag = "required" >
      <module-option name="usersProperties">openCRX.users.properties</module-option>
      <module-option name="rolesProperties">openCRX.roles.properties</module-option>
    </login-module>
  </authentication>
</application-policy>
```

Then create the files *openCRX.users.properties* and *openCRX.roles.properties* in directory *d:\pgm\jboss-4.0.1\server\default\conf*:

### Example 4-3. openCRX.users.properties with user=password syntax.

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

### Example 4-4. openCRX.roles.properties with user.Roles=role1,role2 syntax.

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

Add additional users of your choice to the files.

## Configuring DatabaseServerLoginModule

*openCRX* stores security information in the database tables *security\_Policy*, *security\_Principal*, *security\_Credential*, *security\_Subject* and *security\_Role*. *JBoss* allows to access these tables by configuring a 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 **root** servlet. This simplifies the *openCRX* bootstrapping.

Activate JAAS based authentication by adding the following configuration entries for the **root** servlet to *d:\pgm\jboss-4.0.1\server\default\conf\login\_config.xml* (*login-config.xml* on Unix platforms!).

### Example 4-5. JBoss login\_config.xml for JAAS login configuration for the root servlet.

```
<application-policy name = "opencrx-core-CRX-Root">
  <authentication>
    <login-module code="org.jboss.security.auth.spi.UsersRolesLoginModule" flag = "required" >
      <module-option name="usersProperties">openCRX.users.properties</module-option>
      <module-option name="rolesProperties">openCRX.roles.properties</module-option>
    </login-module>
  </authentication>
</application-policy>
```

Then create the files *openCRX.users.properties* and *openCRX.roles.properties* in directory *d:\pgm\jboss-4.0.1\server\default\conf*:

### Example 4-6. openCRX.users.properties with user=password syntax.

```
admin-Root=rootSecret
```

### Example 4-7. openCRX.roles.properties with user.Roles=role1,role2 syntax.

```
admin-Root.Roles=OpenCrxRoot
```

Add the following security policy for the **standard** servlet for **NON-PostgreSQL** databases:



The *opencrx-core.jboss-3-connector.zip* in the *openCRX* core distribution contains sample files which you can copy/paste.

### Example 4-8. JBoss login\_config.xml for JAAS login configuration for the standard servlet for NON-PostgreSQL databases.

```
<application-policy name="opencrx-core-CRX">
  <authentication>
    <login-module code="org.jboss.security.auth.spi.DatabaseServerLoginModule" flag="required">
      <module-option name="dsJndiName">java:/jdbc_opencrx_CRX</module-option>
      <module-option name="principalsQuery">SELECT c.passwd FROM security_Principal p, security_Credential c WHERE
(p.object_rid IN (SELECT object_rid FROM security_REF WHERE c$0='org:openmdx:security:realm1' AND c$1='provider' AND
c$2='CRX' AND c$3='segment' AND c$4='Root' AND c$5='realm' AND c$6='Default' AND c$7='principal' AND n=8)) AND
(p.p$$credential__rid = c.object_rid) AND (p.p$$credential__oid = c.object_oid) AND (p.object_idx = 0) AND
(p.object_oid = ?)</module-option>
      <module-option name="rolesQuery">SELECT pg.p$$granted_role__oid, 'Roles' FROM security_Principal pg,
security_Principal p WHERE (pg.object_rid = p.p$$is_member_of__rid) AND (pg.object_oid = p.p$$is_member_of__oid) AND
(p.object_rid IN (SELECT object_rid FROM security_REF WHERE c$0='org:openmdx:security:realm1' AND c$1='provider' AND
c$2='CRX' AND c$3='segment' AND c$4='Root' AND c$5='realm' AND c$6='Default' AND c$7='principal' AND n=8)) AND
(p.object_oid = ?)</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>
  </authentication>
</application-policy>
```



The *opencrx-core.jboss-3-connector.zip* in the *openCRX* core distribution contains sample files which you can simply copy/paste.

Add the following security policy for the **standard** servlet for **PostgreSQL** databases:

**Example 4-9. JBoss `login_config.xml` for JAAS login configuration for the standard servlet for PostgreSQL databases.**

```
<application-policy name="opencrx-core-CRX">
  <authentication>
    <login-module code="org.jboss.security.auth.spi.DatabaseServerLoginModule" flag="required">
      <module-option name="dsJndiName">java:/jdbc_opencrx_CRX</module-option>
      <module-option name="principalsQuery">SELECT c.passwd FROM security_Principal p, security_Credential c
WHERE (p.object_rid IN (SELECT object_rid FROM security_REF WHERE "c$0"='org:openmdx:security:realm1' AND
"c$1"='provider' AND "c$2"='CRX' AND "c$3"='segment' AND "c$4"='Root' AND "c$5"='realm' AND "c$6"='Default' AND
"c$7"='principal' AND n=8)) AND (p."p$$credential__rid" = c.object_rid) AND (p."p$$credential__oid" = c.object_oid)
AND
(p.object_idx = 0) AND (p.object_oid = ?)</module-option>
      <module-option name="rolesQuery">SELECT pg."p$$granted_role__oid", 'Roles' FROM security_Principal pg,
security_Principal p WHERE (pg.object_rid = p."p$$is_member_of__rid") AND (pg.object_oid = p."p$$is_member_of__oid")
AND
(p.object_rid IN (SELECT object_rid FROM security_REF WHERE "c$0"='org:openmdx:security:realm1' AND
"c$1"='provider' AND "c$2"='CRX' AND "c$3"='segment' AND "c$4"='Root' AND "c$5"='realm' AND "c$6"='Default' AND
"c$7"='principal' AND n=8)) AND (p.object_oid = ?)</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>
  </authentication>
</application-policy>
```

# Chapter 5. Starting JBoss

You are now ready to start *JBoss*. Open a command shell and start `d:\pgm\jboss-4.0.1\bin\run.bat`. You should verify the following lines in the console output:

Verify whether the start options are as configured in *run.bat* described earlier:

## Example 5-1. JBoss start options.

```
=====
JBoss Bootstrap Environment
JBASS_HOME: D:\pgm\jboss-4.0.1\bin\..
JAVA: D:\pgm\j2sdk1.4.2\bin\java
JAVA_OPTS: -Dprogram.name=run.bat -Xms128m -Xmx512m
-Dorg.openmdx.compatibility.base.application.j2ee.domain=apps
-Dorg.openmdx.compatibility.base.application.j2ee.server=server1
-Djava.protocol.handler.pkgs=org.openmdx.kernel.url.protocol
-Dorg.openmdx.log.config.filename=D:\pgm\jboss-4.0.1\server\default\server.log.properties
-Dmail.SSLSocketFactory.class=org.opencrx.kernel.mail.SendMailSSLSocketFactory
CLASSPATH: D:\pgm\j2sdk1.4.2\lib\tools.jar;D:\pgm\jboss-4.0.1\bin\run.jar
=====
```

## Example 5-2. Deployment of database datasource.

```
[jdbc_opencrx_CRX-MAXDB] Bound connection factory for resource adapter for ConnectionManager
'jboss.jca:service=LocalTxCM,name=jdbc_opencrx_CRX-MAXDB to JNDI name 'java:/jdbc_opencrx_CRX-MAXDB'
```

## Example 5-3. Deployment of opencrx-core-CRX-App.ear.

```
11:08:19,297 INFO [EARDeployer] Init J2EE application: file:/D:/pgm/jboss-4.0.1/server/default/deploy/opencrx-core-CRX-App.ear
11:08:26,587 INFO [EjbModule] Deploying opencrx_core_CRX_gateway_mandatory
11:08:26,768 INFO [EjbModule] Deploying opencrx_core_CRX_gateway_noOrNew
11:08:27,068 INFO [EjbModule] Deploying opencrx_core_CRX_kernel_mandatory
11:08:27,198 INFO [EjbModule] Deploying opencrx_core_CRX_security_mandatory
11:08:27,298 INFO [EjbModule] Deploying opencrx_core_CRX_ui_never
11:08:27,429 INFO [EJBDeployer] Deployed:
file:/D:/pgm/jboss-4.0.1/server/default/tmp/deploy/tmp59451opencrx-core-CRX-App.ear-contents/gateway.jar
11:08:28,050 INFO [EJBDeployer] Deployed:
file:/D:/pgm/jboss-4.0.1/server/default/tmp/deploy/tmp59451opencrx-core-CRX-App.ear-contents/kernel.jar
11:08:28,130 INFO [EJBDeployer] Deployed:
file:/D:/pgm/jboss-4.0.1/server/default/tmp/deploy/tmp59451opencrx-core-CRX-App.ear-contents/security.jar
11:08:28,180 INFO [EJBDeployer] Deployed:
file:/D:/pgm/jboss-4.0.1/server/default/tmp/deploy/tmp59451opencrx-core-CRX-App.ear-contents/ui.jar
11:08:28,210 INFO [TomcatDeployer] deploy, ctxPath=/opencrx-core-CRX/client-gateway, warUrl=
file:/D:/pgm/jboss-4.0.1/server/default/tmp/deploy/tmp59451opencrx-core-CRX-App.ear-contents/client-gateway-exp.war/
11:08:28,430 INFO [EARDeployer] Started J2EE application:
file:/D:/pgm/jboss-4.0.1/server/default/deploy/opencrx-core-CRX-App.ear
```

## Example 5-4. Deployment of opencrx-core-CRX-web.ear and opencrx-core-CRX-Root-web.ear.

```
11:08:28,430 INFO [EARDeployer] Init J2EE application: file:/D:/pgm/jboss-4.0.1/server/default/deploy/opencrx-core-CRX-Root-web.ear/
11:08:35,110 INFO [TomcatDeployer] deploy, ctxPath=/opencrx-core-CRX-Root, warUrl=
file:/D:/pgm/jboss-4.0.1/server/default/deploy/opencrx-core-CRX-Root-web.ear/opencrx-core-CRX-Root.war/
11:08:36,021 INFO [EARDeployer] Started J2EE application: file:/D:/pgm/jboss-4.0.1/server/default/deploy/opencrx-core-CRX-Root-web.ear/
11:08:36,021 INFO [EARDeployer] Init J2EE application: file:/D:/pgm/jboss-4.0.1/server/default/deploy/opencrx-core-CRX-web.ear/
11:08:42,420 INFO [TomcatDeployer] deploy, ctxPath=/opencrx-core-CRX, warUrl=
file:/D:/pgm/jboss-4.0.1/server/default/deploy/opencrx-core-CRX-web.ear/opencrx-core-CRX.war/
11:08:43,251 INFO [EARDeployer] Started J2EE application: file:/D:/pgm/jboss-4.0.1/server/default/deploy/opencrx-core-CRX-web.ear/
```

If there are any errors you must first fix them before you proceed. For troubleshooting please also refer to *here* (<http://www.opencrx.org/faq.htm>).

## Example 5-5. JBoss running – last few lines of console output.

```
11:08:43,492 INFO [Http11Protocol] Starting Coyote HTTP/1.1 on http-0.0.0.0-8080
11:08:44,363 INFO [ChannelSocket] JK2: ajp13 listening on /0.0.0.0:8009
11:08:44,363 INFO [JKMain] Jk running ID=0 time=0/30 config=null
11:08:44,403 INFO [Server] JBoss (MX MicroKernel) [4.0.1 (build: CVSTag=JBoss_4_0_1 date=200412230944)] Started in 38s:529ms
```

Now you are ready to continue with the openCRX QuickStart Guide or you can *Install openCRX as Windows Service*.

## Chapter 6. Install openCRX as Windows Service

If you want to install *JBoss / openCRX* on a Windows platform as a Windows service you can do this as follows:

- Download *JavaService* from *here* (<http://www.opencrx.org/downloads/JavaService-bin-1.2.0.zip>).
- Copy *JavaService.exe* to **D:\pgm\jboss-4.0.1\server\default\JBossDefault.exe**.
- Execute the following command:

### Example 6-1. Starting JBoss as Windows service.

```
JBossDefault.exe -install JBossDefault d:\pgm\j2sdk1.4.2\jre\bin\server\jvm.dll
-Dorg.openmdx.compatibility.base.application.j2ee.domain=apps
-Dorg.openmdx.compatibility.base.application.j2ee.server=server1
-Djava.protocol.handler.pkgs=org.openmdx.kernel.url.protocol
-Dorg.openmdx.log.config.filename=D:\pgm\jboss-4.0.1\server\default\server.log.properties
-Dmail.SSLSocketFactory.class=org.opencrx.kernel.mail.SendMailSSLSocketFactory
-Dprogram.name=run.bat
-Djava.class.path=d:\pgm\j2sdk1.4.2\lib\tools.jar;D:\pgm\jboss-4.0.1\bin\run.jar
-Xms128m -Xmx500m -start org.jboss.Main
```



Adapt **d:\pgm\j2sdk1.4.2** and **D:\pgm\jboss-4.0.1** to your environment!

# Chapter 7. Install JBoss as daemon and Startup Service on Linux

This section (provided by Seah Hong Yee) is devoted to the automatic start up of jboss services during the startup phase of a server. It also simplifies manual jboss startup with the use of System V init script. The following configuration has been tested on Mandrake Linux 10.1 and SuSe Linux Enterprise Server 9. Based on the particular distribution at hand there might be some minor differences in init scripts and configuration, but the following guide should work with RHEL, CentOS, WhiteBox and Fedora.

In the directory `$JBOSS_HOME/bin` there should be two init scripts:

- `jboss_init_redhat.sh`
- `jboss_init_suse.sh`

If you are using **Mandrake/Mandriva, RHEL, CentOS, WhiteBox or Fedora:**

- Copy the `jboss_init_redhat.sh` script into `/etc/init.d` and rename it to `jboss`.
- Edit the script and adapt the following parameters: `JBOSS_HOME` and `JAVAPATH`
- Although not strictly necessary, you might want to include an entry like:

```
export PATH=/usr/java/j2sdk1.4.2_08/bin
```

(`/usr/java/j2sdk1.4.2_08/` being your jdk path, adapt it to your environment)

- At the top of the script there's an entry that resembles the following one:

```
# chkconfig: 3 87 20
```

The second set of digits represents the order sequence of the service startup. Make sure the number is larger than your database startup. Typically postgresql starts with the sequence number of 85, so I have my jboss startup with the sequence of 87

- Type the commands

```
# chkconfig --add jboss
# chkconfig jboss on
```

From now on jboss should startup automatically after reboot, or you can do it manually with `service jboss restart`.

For **Suse** Linux, please do the following :

- Copy the `jboss_init_suse.sh` script into `/etc/init.d` and rename it to `jboss`.
- Edit the script and change the following parameters: `JBOSS_HOME` and `JAVAPATH`
- Although not strictly necessary, you might want to include an entry near the top of the script:

```
export PATH=/usr/java/j2sdk1.4.2_08/bin
```

(`/usr/java/j2sdk1.4.2_08/` being your jdk path, adapt it to your environment)

- Type the command "`inserv jboss`"
- Go into directory `/usr/sbin` and create a symbolic link with "`ln -s /etc/init.d/jboss rcjboss`"
- Execute the command "`chkconfig jboss on`"

You should be able to start jboss with the command "`rcjboss start`"

# Appendix A. Appendix

# Bibliography

[01] *openCRX - the leading open source CRM solution*, [opencrx.org](http://www.opencrx.org).

@ <http://www.opencrx.org> (<http://www.opencrx.org>)

[02] *openMDX - The leading open source MDA platform*, [openmdx.org](http://www.openmdx.org).

@ <http://www.openmdx.org> (<http://www.openmdx.org>)