

# openCRX Installation Guide for WebLogic 8

**Version 1.9.0**

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

**openCRX Installation Guide for WebLogic 8: Version 1.9.0**

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 WebLogic</b> .....	<b>3</b>
<b>4. Installing Libraries</b> .....	<b>4</b>
<b>5. Configuring the Java Virtual Machine</b> .....	<b>5</b>
<b>6. Configuring the Datasource</b> .....	<b>6</b>
<b>7. Configuring Security</b> .....	<b>14</b>
<b>8. Deploying openCRX</b> .....	<b>18</b>
<b>9. Next Steps</b> .....	<b>22</b>
<b>A. Appendix</b> .....	<b>23</b>
<b>Bibliography</b> .....	<b>24</b>

# List of Figures

3-1. Launch the WebLogic Server Console. ....	3
6-1. Select the JDBC database vendor. ....	6
6-2. Select the JDBC driver.....	7
6-3. Configure the connection pool. ....	8
6-4. Configure the connection pool. ....	9
6-5. Create and deploy the connection pool. ....	10
6-6. Create and configure the datasource. ....	11
6-7. Select the connection pool. ....	12
6-8. Select the deployment targets for the datasource. ....	13
7-1. Create user groups.....	14
7-2. Create global security roles.....	15
7-3. Add member of the group condition to global role. ....	16
7-4. Create user and make member of group.....	17
8-1. Upload the opencrx-core-CRX-App application enterprise archive. ....	18
8-2. Deploy the opencrx-core-CRX-App.ear enterprise archive. ....	19
8-3. Select the deployment target and assign name. ....	20
8-4. Verify the installation. ....	21

# List of Examples

5-1. openCRX-specific Java VM properties .....	5
--	---

# 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 *WebLogic 8.x*. The intended audience are *openCRX* and application server system administrators.

## What do you need to understand this book

The book assumes that you are familiar with *WebLogic* 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* and *WebLogic*.

- Purchase/Install **WebLogic 8.x** (please note that we have tested openCRX on WebLogic 8.1.x)
- 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 for *jre-1.4* (e.g. *opencrx-1.9.0-core.CRX.jre-1.4.zip*).

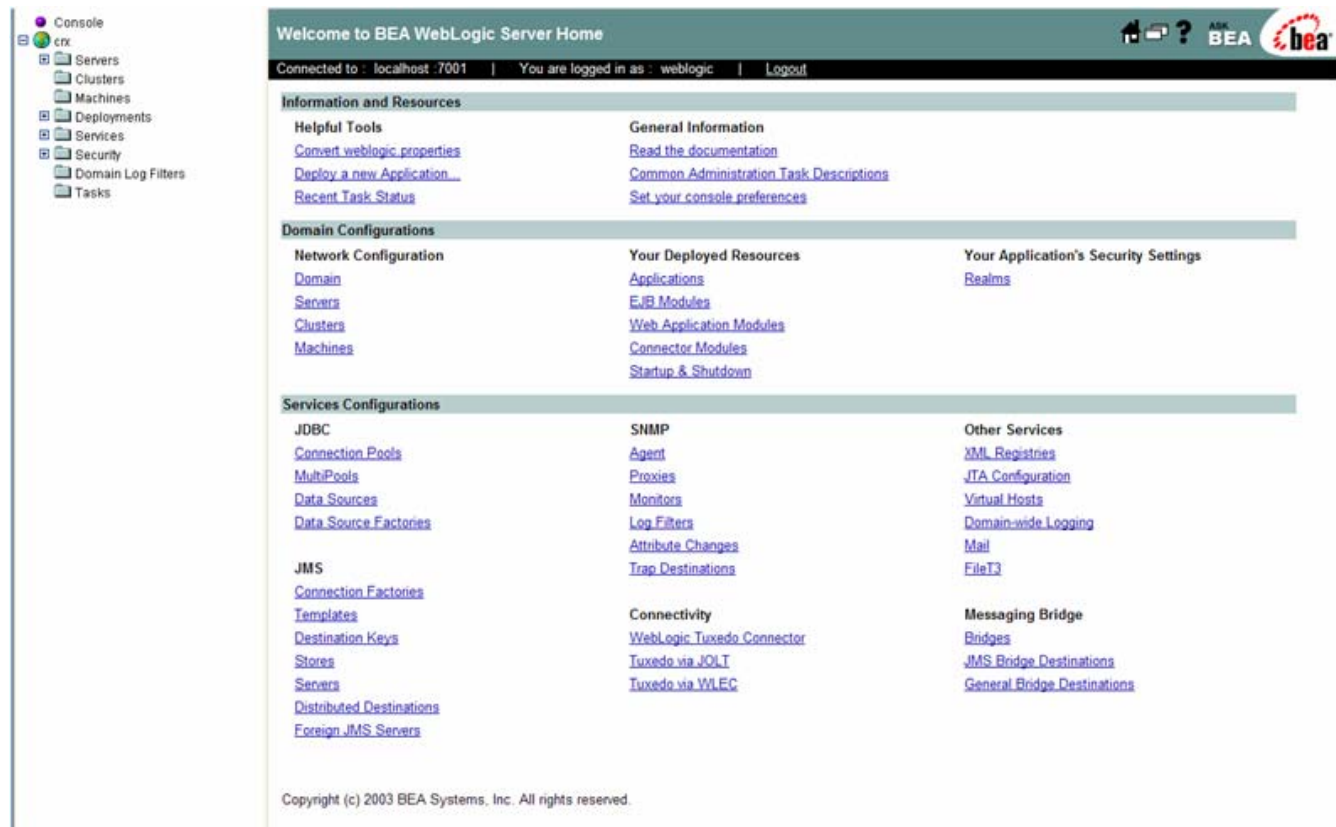


**Important** As a first step you must install the database as described in the database distribution. E.g. if you want to install *openCRX* for *Oracle* you must first install *Oracle* and the corresponding *openCRX* database schema. Similarly, if you want to install *openCRX* for *MS SQL* you must first install *MS SQL* and the corresponding *openCRX* database schema. If you have successfully installed the database you are ready to continue with the *WebLogic* setup.

# Chapter 3. Installing openCRX for WebLogic

In a first step you must install *WebLogic*. You should be able to start and stop *WebLogic* and launch the *WebLogic Server Console* as shown in *Figure 3-1*.

**Figure 3-1. Launch the WebLogic Server Console.**



The *openCRX* installation requires the following steps:

- Configure properties for the Java Virtual Machine required by *openCRX* and *openMDX*.
- Create and configure the datasource to access the *openCRX* database.
- Configure security, i.e. global roles, groups and principals.
- Deploy the *openCRX* application enterprise archives.
- Start and test *openCRX*

# Chapter 4. Installing Libraries

As a first step you must copy the *openMDX* and JDBC driver libraries to a place where *WebLogic* has access to it. You can do this as follows:

- Create a folder where you can store the libraries and *WebLogic* has access to it. E.g. create the directory `..\bea\user_projects\domains\mydomain\lib`.
- **openMDX.** Copy the library *openmdx-kernel.jar* to this newly created directory.
- **Database.** Copy the Jdbc libraries for your database to this newly created directory. E.g. for *Microsoft SQL Server 2000* the files are *msbase.jar*, *mssqlserver.jar*, *msutil.jar*

# Chapter 5. Configuring the Java Virtual Machine

Second you must add some options to the Java Virtual Machine configuration. You can do this by adding the following settings to the *WebLogic* startup script, e.g. *startWebLogic.cmd* as follows:

## Example 5-1. openCRX-specific Java VM properties

```
rem Setup openCRX-specific properties
set DOMAIN_HOME=c:\pgm\bea\user_projects\domains\mydomain
set CLASSPATH=%CLASSPATH%;%DOMAIN_HOME%\lib\openmdx-kernel.jar
set CLASSPATH=%CLASSPATH%;%DOMAIN_HOME%\lib\msutil.jar;%
set CLASSPATH=%CLASSPATH%;%DOMAIN_HOME%\lib\msbase.jar;
set CLASSPATH=%CLASSPATH%;%DOMAIN_HOME%\lib\mssqlserver.jar
set JAVA_OPTIONS=%JAVA_OPTIONS% -Dorg.openmdx.compatibility.base.application.j2ee.domain=apps
set JAVA_OPTIONS=%JAVA_OPTIONS% -Dorg.openmdx.compatibility.base.application.j2ee.server=server1
set JAVA_OPTIONS=%JAVA_OPTIONS% -Djava.protocol.handler.pkgs=org.openmdx.kernel.url.protocol
set JAVA_OPTIONS=%JAVA_OPTIONS% -Dorg.openmdx.log.config.filename=%DOMAIN_HOME%\server.log.properties
set JAVA_OPTIONS=%JAVA_OPTIONS% -Dmail.SSLSocketFactory.class=org.opencrx.ssl.DefaultSSLSocketFactory

@REM Call WebLogic Server
...
```



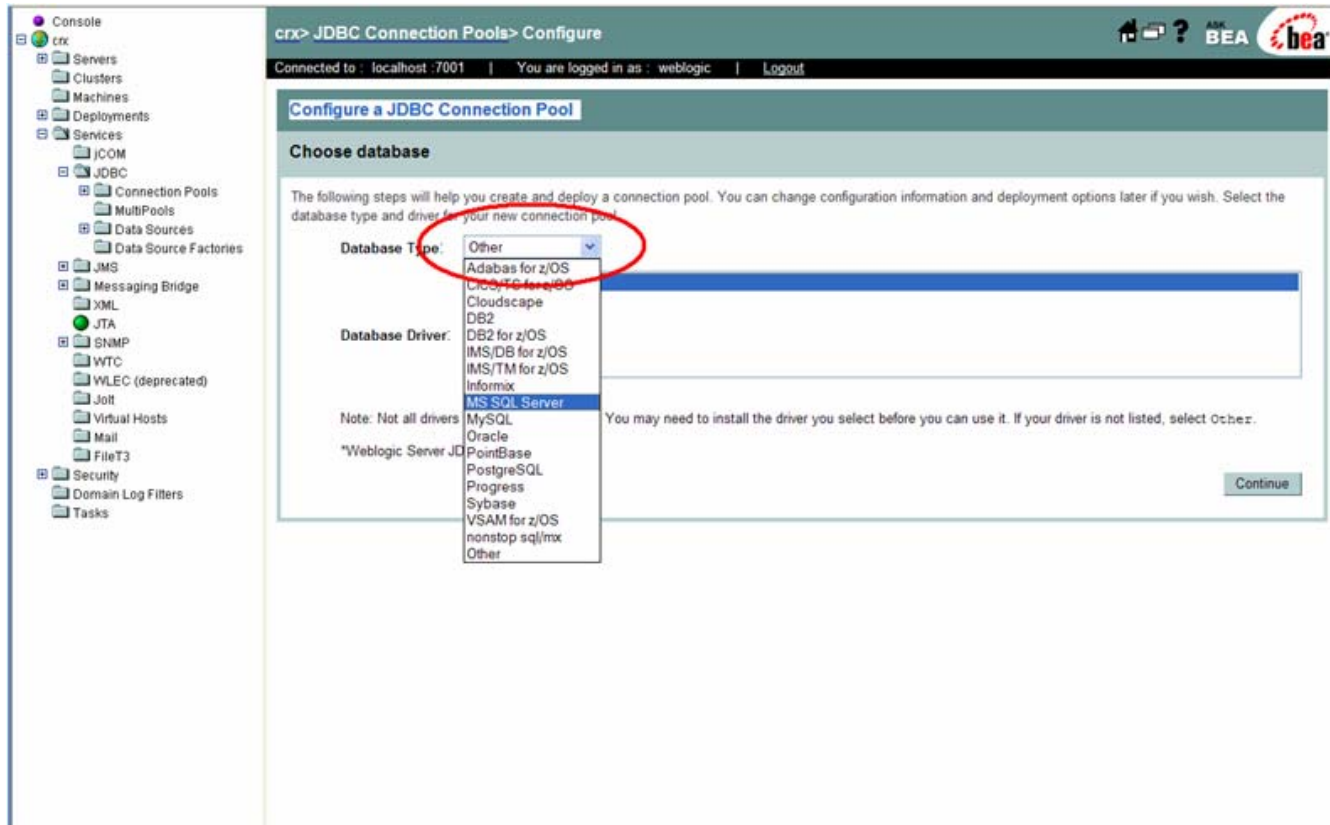
Before you continue you must restart (stop and start) *WebLogic*. The newly configured libraries and environment variables are not active until you restart *WebLogic*.

# Chapter 6. Configuring the Datasource

The *openCRX* application requires the configuration of a JDBC datasource which connects to the *openCRX* database. You can do this as follows:

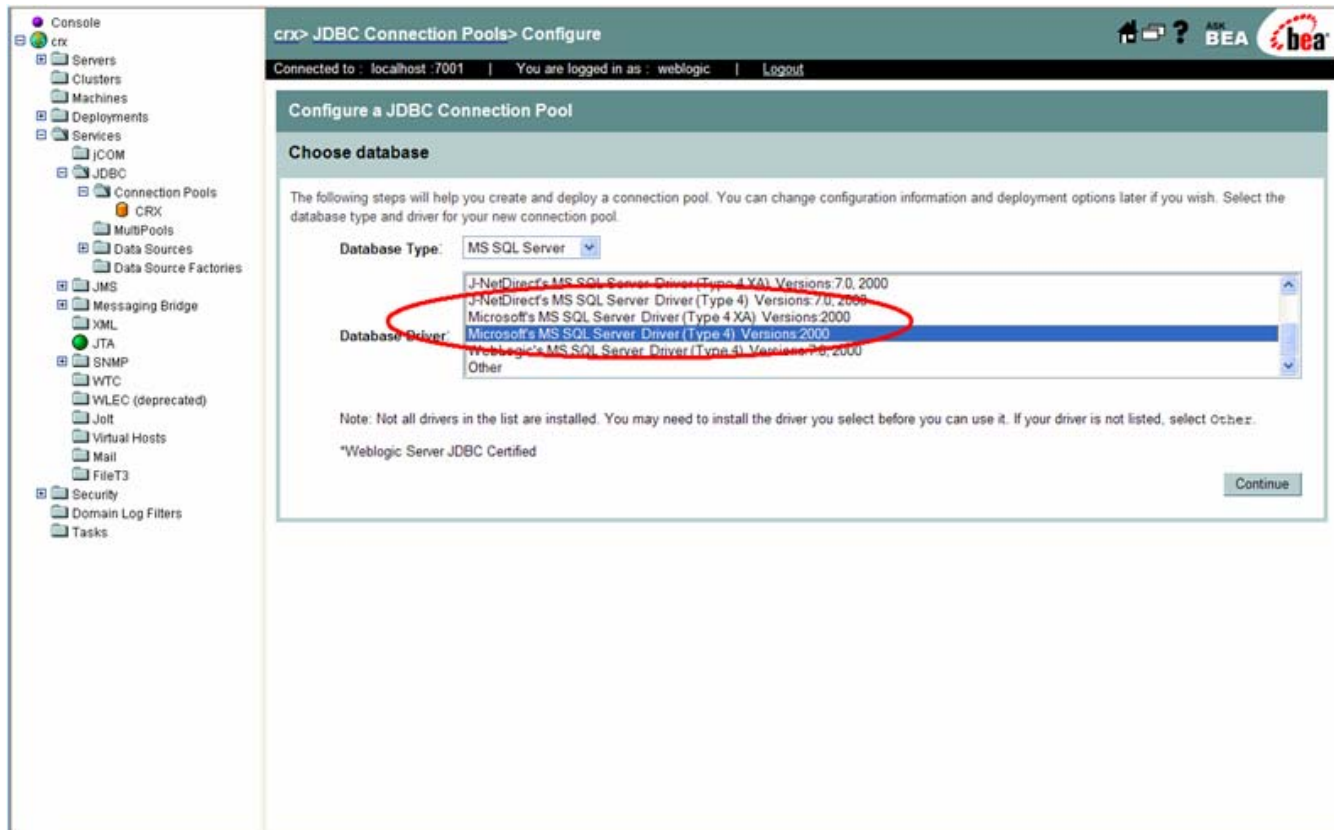
Navigate to *Services > JDBC > Connection Pools* and click on *Configure a new JDBC Connection Pool*. Select the database vendor from the drop down, e.g. *MS SQL Server* as shown in *Figure 6-1*.

**Figure 6-1. Select the JDBC database vendor.**



Then select the driver from the *Database Driver* drop down, e.g. *Microsoft's MS SQL Server Driver (Type) Version 2000* as shown in *Figure 6-2*. Then click continue.

**Figure 6-2.** Select the JDBC driver.



On the next form you must configure the connection pool. You must set the following properties:

- **Name.** Enter the name of the connection pool *crx-CRX*.
- **Database Name.** The name must match the name of your installed database, typically *crx-CRX*.
- **Host Name.** Enter the host name of your database server, e.g. *localhost*.
- **Port.** Enter the port number which is used for your database service, e.g. for *Microsoft SQL Server 2000* this is typically *1433*.
- **Database User Name / Password.** Enter the user name and password which is used as database login.

This is shown in *Figure 6-3*.

**Figure 6-3. Configure the connection pool.**

The screenshot shows the 'Configure a JDBC Connection Pool' web interface. The form is titled 'Configure a JDBC Connection Pool' and 'Define connection properties'. It contains several input fields: 'Name' (crx-CRX), 'Database Name' (crx-CRX), 'Host Name' (localhost), 'Port' (1433), 'Database User Name' (system), 'Password' (masked with dots), and 'Confirm Password' (masked with dots). Each of these fields is circled in red. A 'Continue' button is located at the bottom right of the form. The left sidebar shows a navigation tree with 'JDBC' and 'Connection Pools' expanded.

Then click continue.

Verify the database connection properties on the next page as shown in *Figure 6-4* and then click *Test Driver Configuration*.

**Figure 6-4. Configure the connection pool.**

The screenshot displays the 'Configure a JDBC Connection Pool' window in the Oracle JDeveloper console. The left-hand navigation pane shows the project structure, with 'JDBC' expanded to 'Connection Pools'. The main window title is 'crx> JDBC Connection Pools> Configure'. Below the title bar, it indicates 'Connected to : localhost:7001' and 'You are logged in as : weblogic'. The 'Test database connection' section is highlighted, with a sub-header 'Test database connection'. A descriptive paragraph states: 'A connection pool can fail if the database cannot be reached or if there are errors in the connection properties. Ping your database now to test database availability and the connection properties you provided.' The configuration fields are as follows:

- Driver Classname:** `com.microsoft.jdbc.sqlserver.SQLServerDriver`
- URL:** `jdbc:microsoft:sqlserver://localhost:1433`
- Database User Name:** `system`
- Password:** (masked with dots)
- Confirm Password:** (masked with dots)
- Properties:**

```

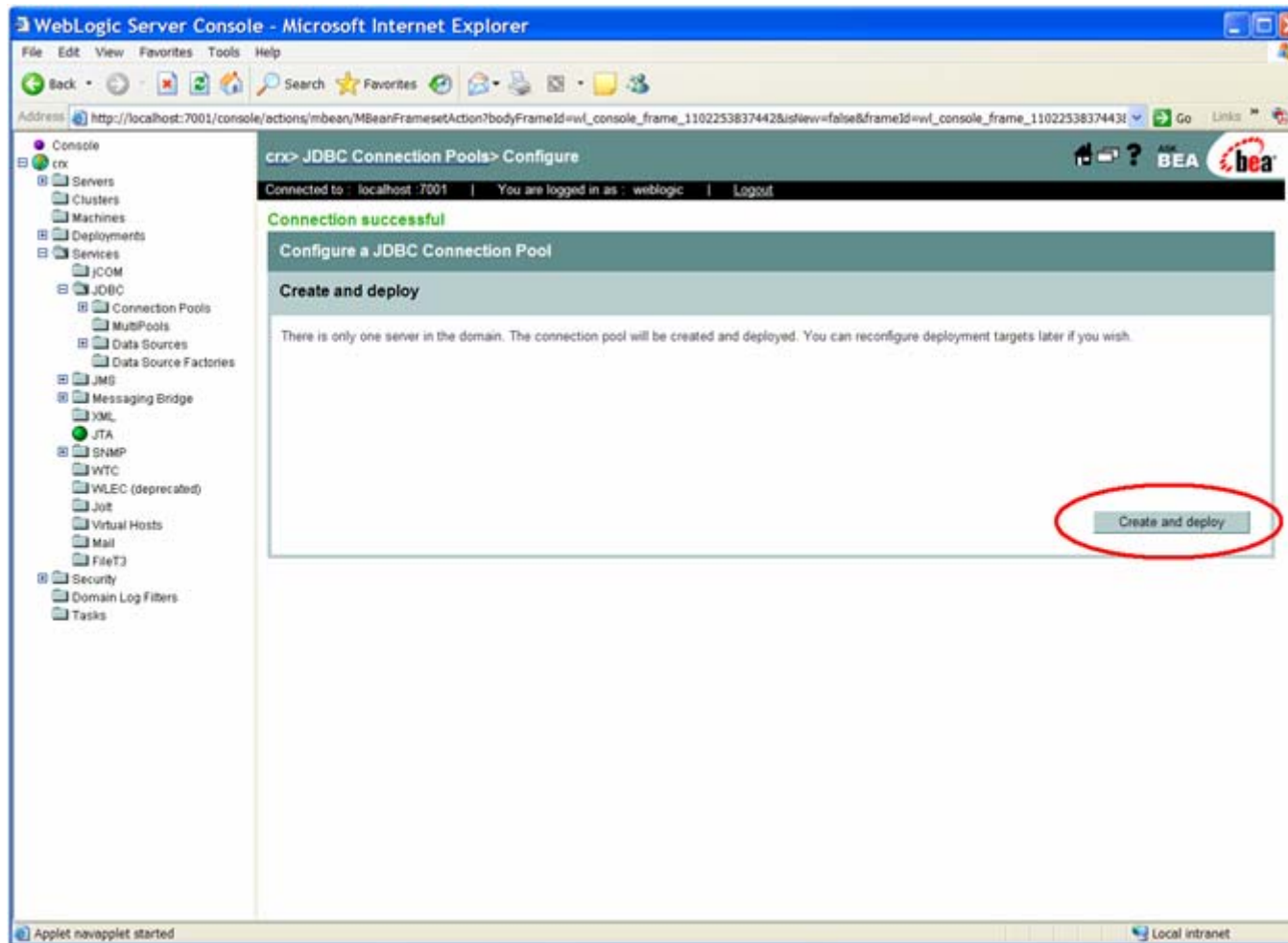
user=system
url=jdbc:microsoft:sqlserver://localhost:1433
selectMethod=cursor
dataSourceName=SQL2000JDBC

```

At the bottom of the dialog, there are two buttons: 'Test Driver Configuration' (which is circled in red) and 'Skip this Step'.

If everything works fine you should see the *Connection successful* message as shown in *Figure 6-5*. Click *Create and deploy*. This finally creates the connection pool.

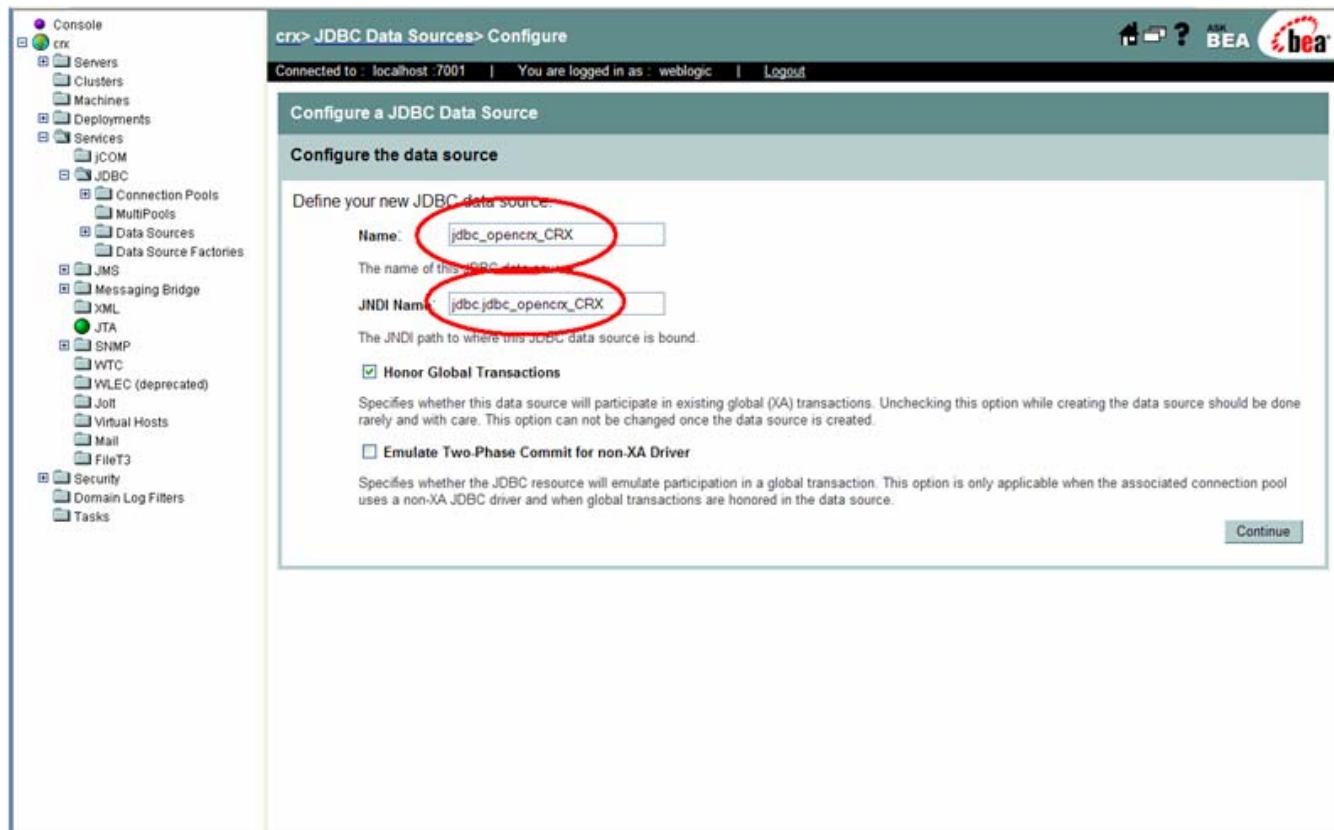
**Figure 6-5. Create and deploy the connection pool.**



As next step you must create a datasource using the connection pool you have just created. Navigate to *Services > JDBC > Data Sources* and click *Configure a new JDBC Data Source* as shown in Figure 6-6. You must set the properties as follows:

- **Name.** *jdbc\_opencrx\_CRX*.
- **JNDI Name.** *jdbc.jdbc\_opencrx\_CRX*. It is very important that you set the JNDI correctly. The datasource is referenced from the *openCRX* application by this name.

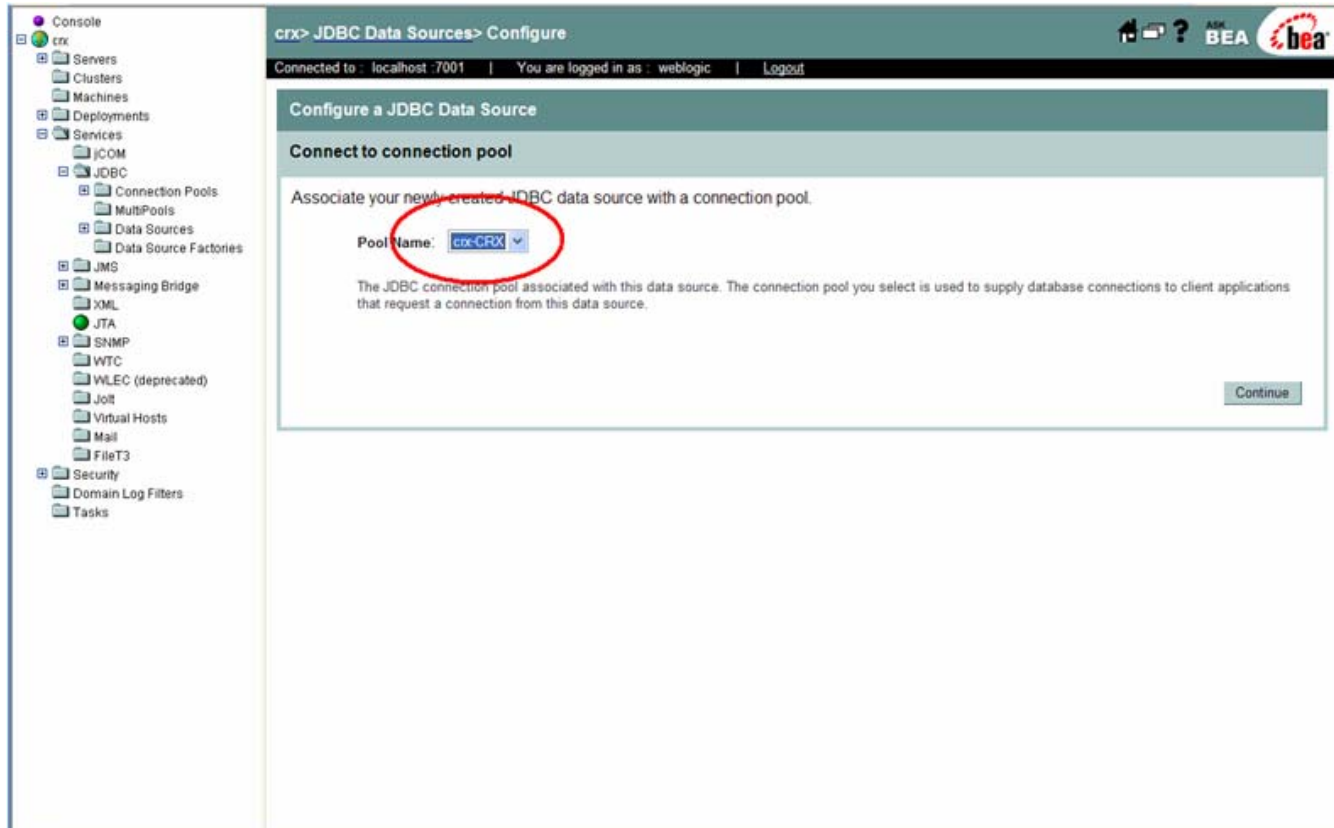
**Figure 6-6. Create and configure the datasource.**



Then click *Continue*.

On the next page you must select the database connection pool from the drop down. Select the pool you have just created, i.e. *crx-CRX* as shown in *Figure 6-7*.

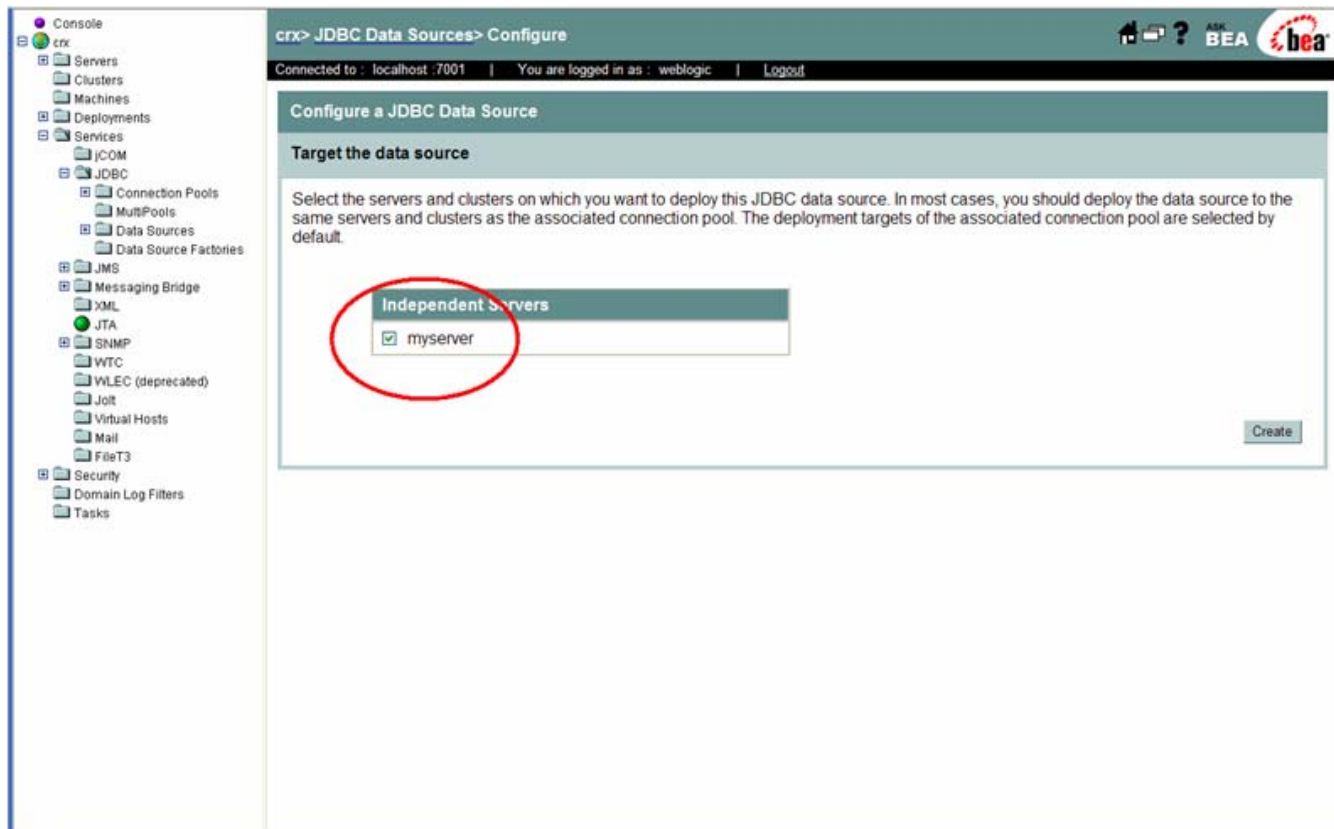
**Figure 6-7. Select the connection pool.**



Then click *Continue*.

On the next page you must select the servers where the new datasource should be deployed to. You must select the servers to which you plan to deploy the *openCRX* application. This is shown in *Figure 6-8*.

**Figure 6-8.** Select the deployment targets for the datasource.



Then click *Create*. You are now finished creating the datasource for *openCRX*.

# Chapter 7. Configuring Security

*openCRX* requires that each user is properly authenticated. This allows *openCRX* to correlate a session to user-specific application data and to perform access control. *openCRX* does not support non-authenticated sessions. User authentication must be configured as follows:

- Select *Security > Realms > myrealm*.
- Then click *Groups > Configure a new Group*. You must create the following groups: *OpenCrxUser*, *OpenCrxAdministrator*, *OpenCrxRoot* as shown in *Figure 7-1*.

**Figure 7-1. Create user groups.**

The screenshot shows the BEA WebLogic Administration Console interface. On the left is a navigation tree with the following structure:

- Console
- crx
  - Servers
  - Clusters
  - Machines
  - Deployments
  - Services
  - Security
    - Realms
      - myrealm
        - Users
        - Groups
        - Global Roles
      - Providers
    - Domain Log Filters
    - Tasks

The main content area is titled "Groups" and shows the configuration for the "myrealm" security realm. It includes a "Filter By:" field and a "Filter" button. Below this is a table of groups:

Group	Description	Provider
<a href="#">Administrators</a>	Administrators can view and modify all resource attributes and start and stop servers.	DefaultAuthenticator
<a href="#">Deployers</a>	Deployers can view all resource attributes and deploy applications.	DefaultAuthenticator
<a href="#">Monitors</a>	Monitors can view and modify all resource attributes and perform operations not restricted by roles.	DefaultAuthenticator
<a href="#">Operators</a>	Operators can view and modify all resource attributes and perform server lifecycle operations.	DefaultAuthenticator
<a href="#">OpenCrxUser</a>		DefaultAuthenticator
<a href="#">OpenCrxAdministrator</a>		DefaultAuthenticator
<a href="#">OpenCrxRoot</a>		DefaultAuthenticator

The groups *OpenCrxUser*, *OpenCrxAdministrator*, and *OpenCrxRoot* are circled in red in the original image.

Then select *Global Roles > Configure a new Global Role*. Create the roles *OpenCrxRoot*, *OpenCrxAdministrator* and *OpenCrxUser* as shown in *Figure 7-2*.

**Figure 7-2. Create global security roles.**

The screenshot shows the BEA WebLogic console interface. On the left is a navigation tree with 'Global Roles' selected. The main content area is titled 'Global Roles' and includes a status bar showing 'Connected to: localhost:7001' and 'You are logged in as: weblogic'. Below this, there is explanatory text about security roles and global roles. A link 'Configure a new Global Role...' is visible. A table lists the configured global roles, with three roles circled in red: 'OpenCrxAdministrator', 'OpenCrxRoot', and 'OpenCrxUser'.

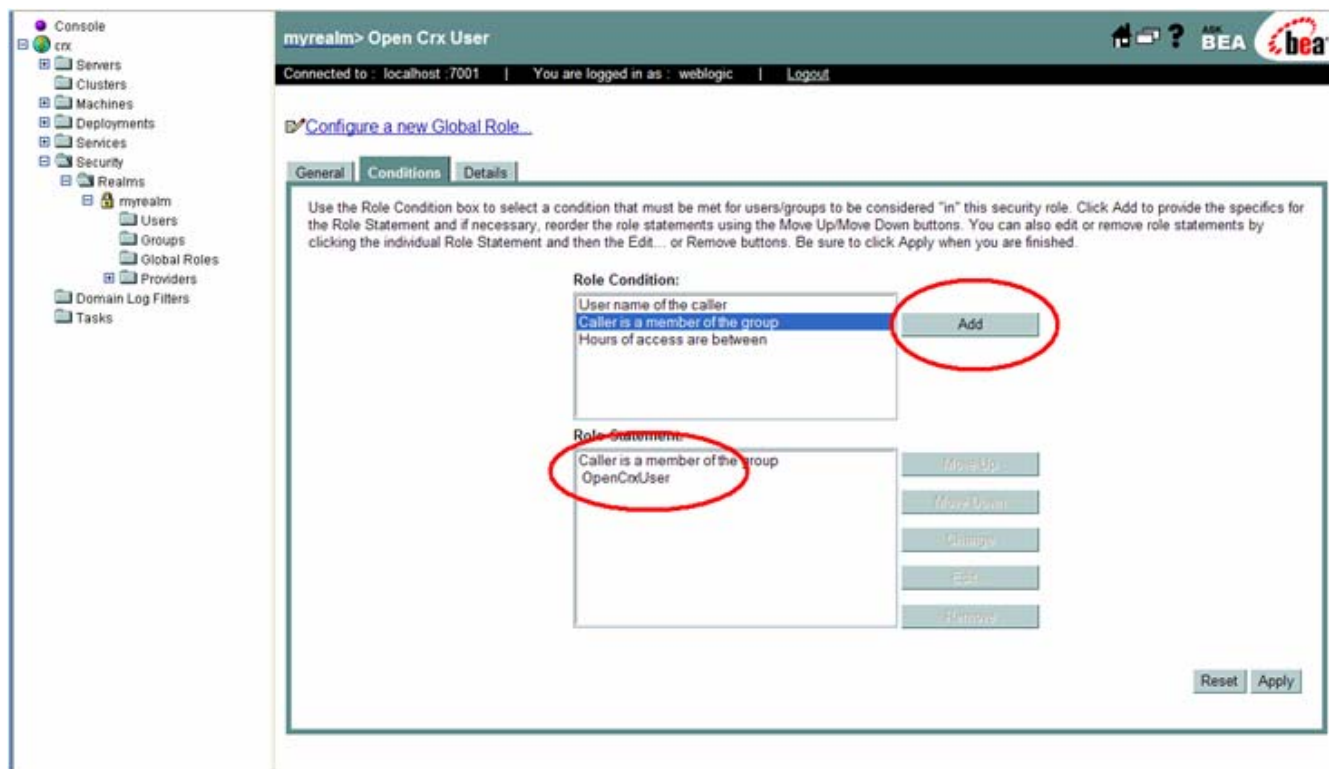
Global Role	Provider	
<a href="#">Admin</a>	DefaultRoleMapper	
<a href="#">Anonymous</a>	DefaultRoleMapper	
<a href="#">Deployer</a>	DefaultRoleMapper	
<a href="#">monitor</a>	DefaultRoleMapper	
<a href="#">OpenCrxAdministrator</a>	DefaultRoleMapper	
<a href="#">OpenCrxRoot</a>	DefaultRoleMapper	
<a href="#">OpenCrxUser</a>	DefaultRoleMapper	
<a href="#">Operator</a>	DefaultRoleMapper	

As a next step you must add the *Caller is a member of the group* condition to the newly created global roles as follows:

- Select the global role *OpenCrxUser* and select the tab *Conditions*. In the *Role Condition* pane select the entry *Caller is a member of the group* and then click *Add*. Enter the group name *OpenCrxUser* and save the modifications with *Apply*.
- Select the global role *OpenCrxAdministrator* and select the tab *Conditions*. In the *Role Condition* pane select the entry *Caller is a member of the group* and then click *Add*. Enter the group name *OpenCrxAdministrator* and save the modifications with *Apply*.
- Select the global role *OpenCrxRoot* and select the tab *Conditions*. In the *Role Condition* pane select the entry *Caller is a member of the group* and then click *Add*. Enter the group name *OpenCrxRoot* and save the modifications with *Apply*.

This is shown in *Figure 7-3*.

**Figure 7-3. Add member of the group condition to global role.**

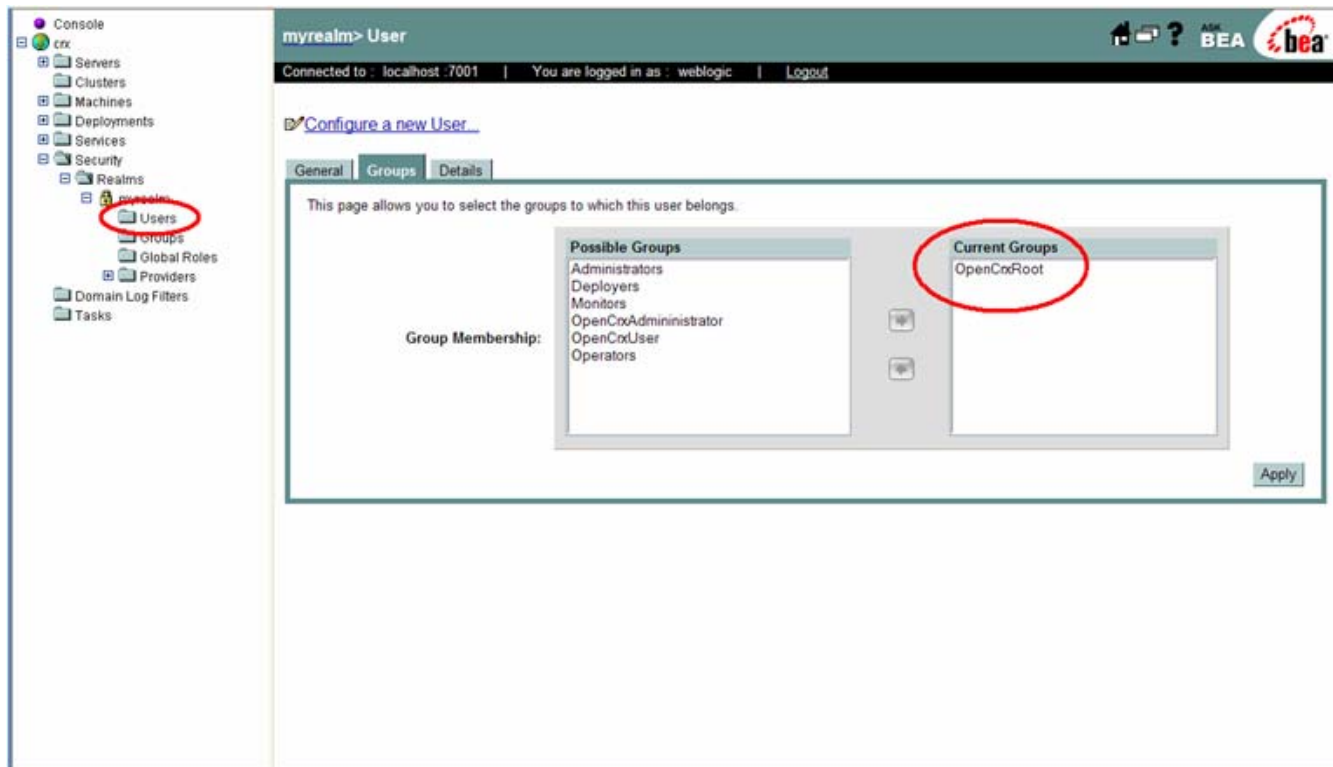


You now have successfully mapped the role names defined by the *openCRX* application to *WebLogic*-defined role names.

You are now ready to create users and make them member of the groups *OpenCrxRoot*, *OpenCrxAdministrator* or *OpenCrxUser*. A user must be member of *OpenCrxRoot* if he/she wants to access the servlet *opencrx-core-CRX-Root*. The servlet *opencrx-core-core-CRX* requires to be member of the groups *OpenCrxUser* or *OpenCrxAdministrator*. Standard users are member of the group *OpenCrxUser*. Segment administrators are member of the group *OpenCrxAdministrator*. This is shown in *Figure 7-4*. As a start you can add the following users:

- **admin-Root:** root user which is allowed to access the Root servlet. Make it member of the group *OpenCrxRoot*.
- **admin-Standard:** administrator for segment standard. Make the user member of the group *OpenCrxAdministrator*.
- **user1:** test user. Make it member of *OpenCrxUser*.

**Figure 7-4. Create user and make member of group.**



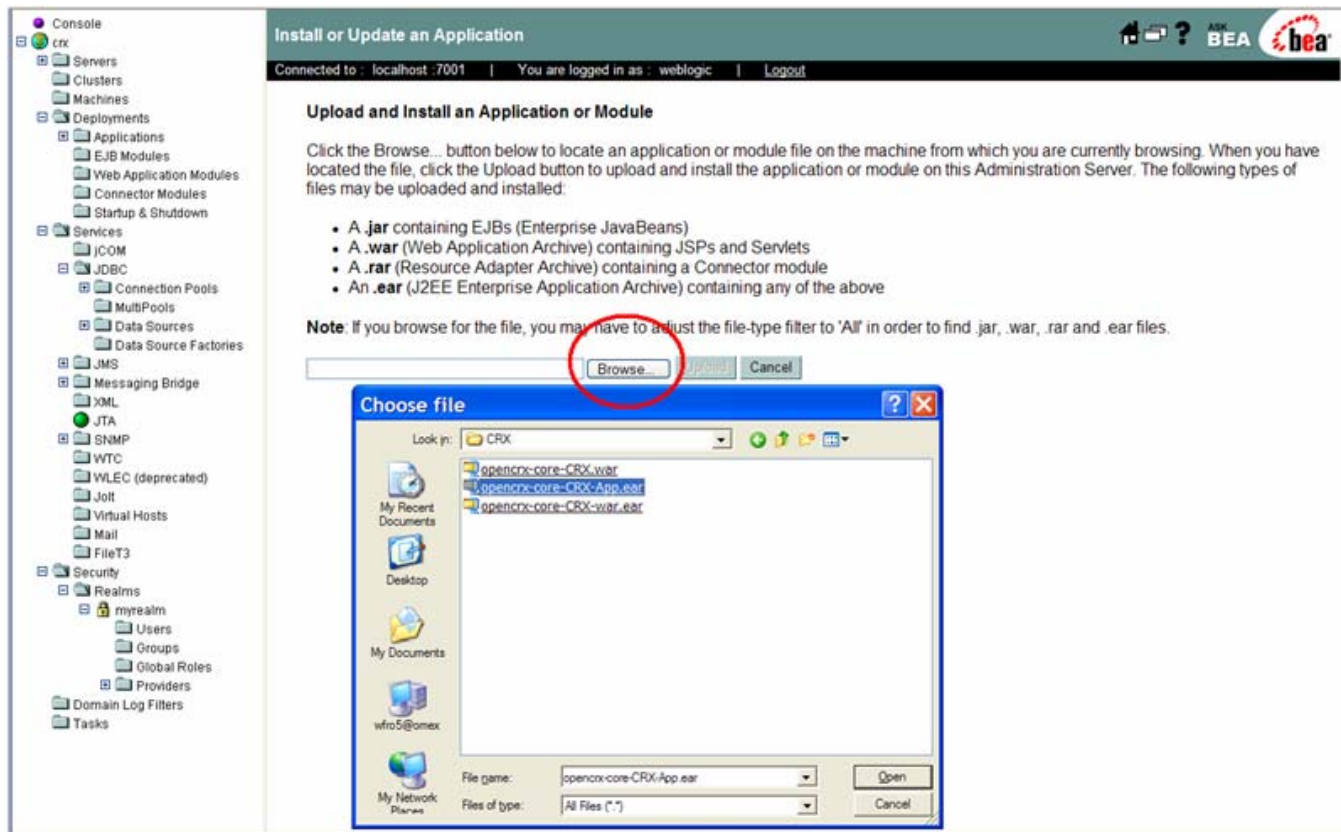
# Chapter 8. Deploying openCRX

*openCRX* comes with two enterprise application archives (EAR):

- **opencrx-core-CRX-App.ear**. contains the *openCRX* server components, i.e. Enterprise Java Beans.
- **opencrx-core-CRX-web.ear**. contains the web application for standard *openCRX* users.

In a first step you deploy *opencrx-core-CRX-App.ear*. Select *Deployments > Applications > Deploy a new Application > Upload your files*. Click *Browse* and select the file *opencrx-core-CRX-App.ear* as shown in *Figure 8-1*.

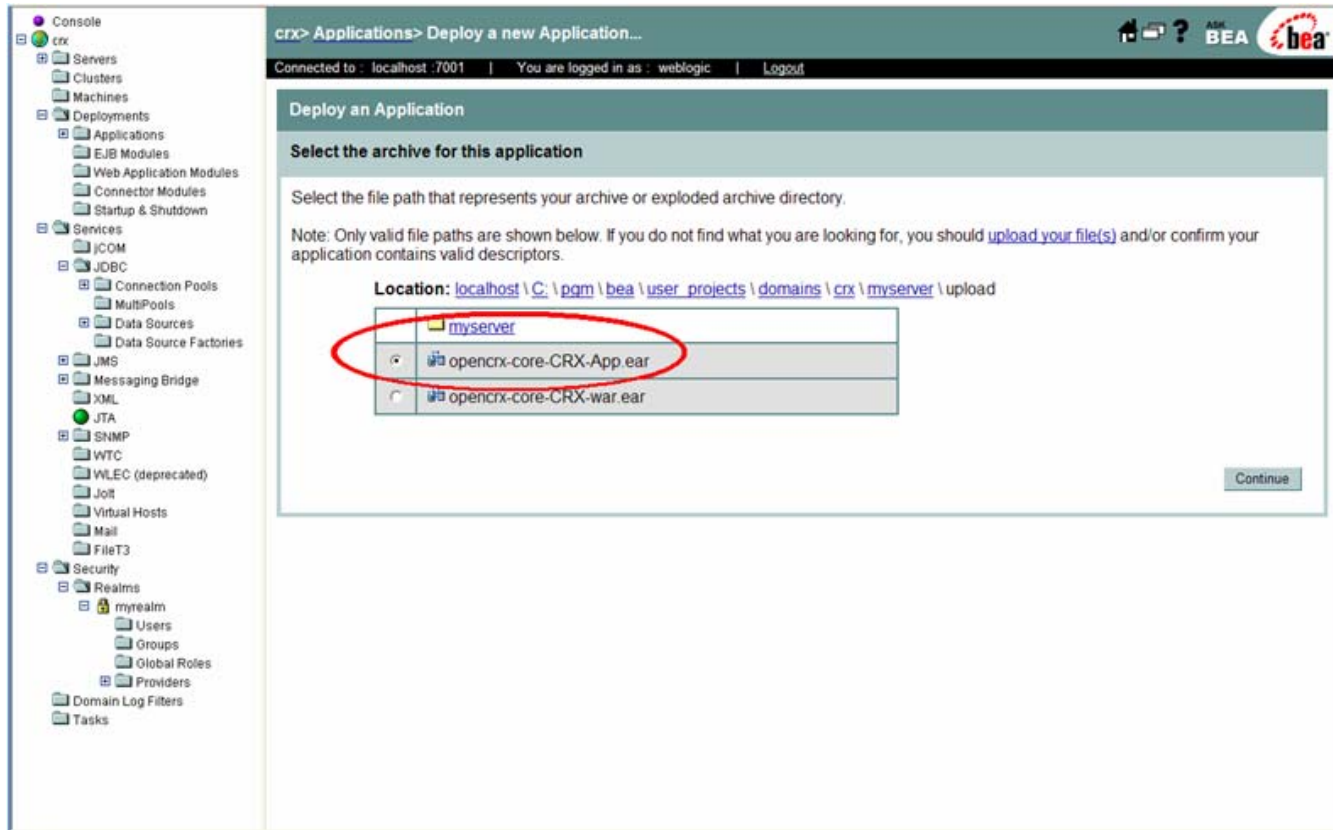
**Figure 8-1. Upload the opencrx-core-CRX-App application enterprise archive.**



Repeat this step for the file *opencrx-core-CRX-web.ear*. The files are now ready for deployment.

Navigate now to *Deployments > Applications > Deploy a new Application* and then to the upload directory of your administration server, e.g. *domains/mydomain/myserver/upload*. In this directory you should find the files you have just uploaded. Select the file *opencrx-core-CRX-App.ear* as shown in *Figure 8-2*.

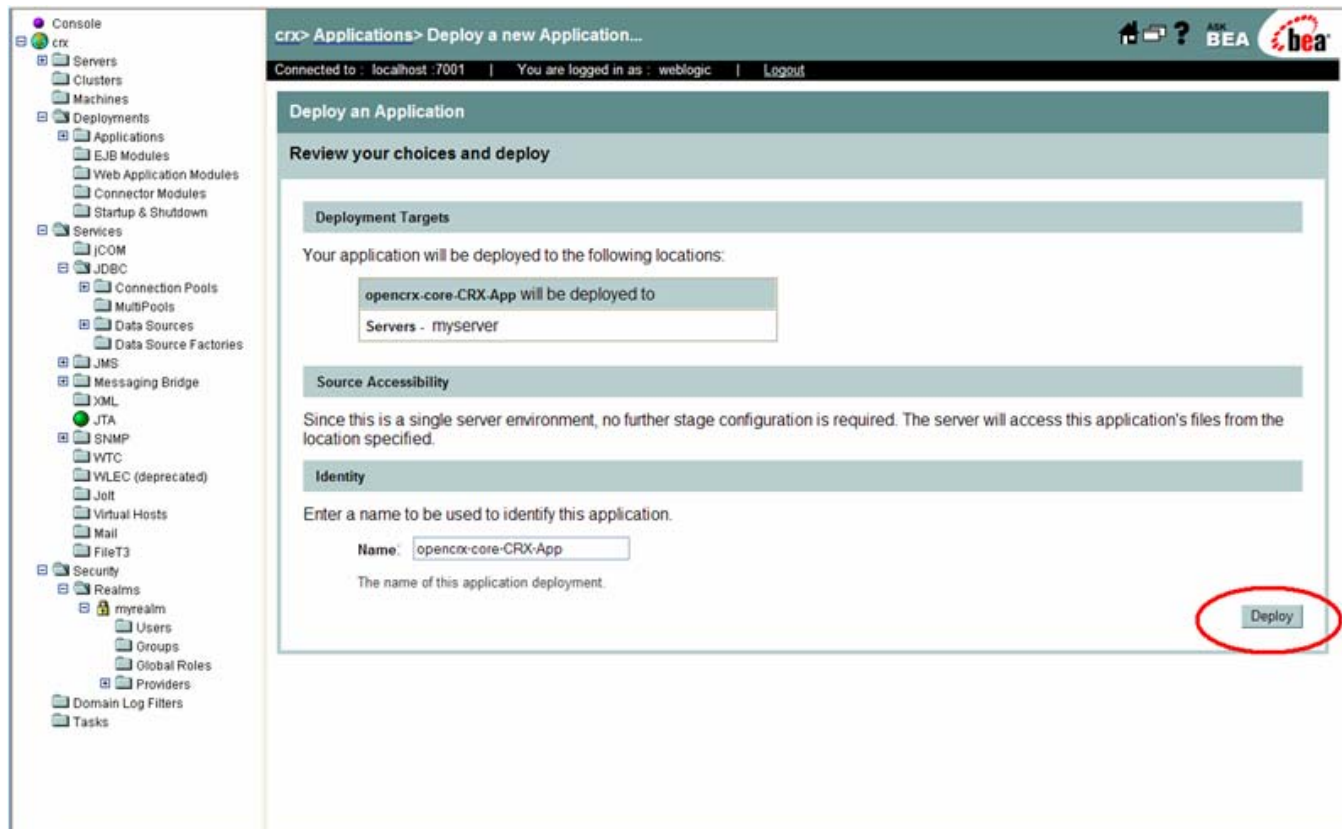
**Figure 8-2.** Deploy the *opencrx-core-CRX-App.ear* enterprise archive.



Then click *Continue*.

On the next screen you must specify the deployment targets and the application name. Leave the default values unchanged and then click *Deploy* as shown in *Figure 8-3*.

**Figure 8-3.** Select the deployment target and assign name.



Repeat the deployment step for the archive *opencrx-core-CRX-web.ear*.

Finally, the tree of the deployed applications should look as shown in *Figure 8-4*.

**Figure 8-4. Verify the installation.**

The screenshot displays the BEA WebLogic Administration Console interface. On the left, a tree view shows the 'Applications' folder expanded, with 'opencrx-core-CRX-App' and 'opencrx-core-CRX-war' highlighted. The main content area shows the 'Applications' page with a table of deployed applications. The table has columns for Name, Path, Deployment Order, and Modules. Two rows are visible, both with a deployment order of 100. The first row is 'opencrx-core-CRX-App' with 5 modules, and the second row is 'opencrx-core-CRX-war' with 1 module. Red circles highlight the application names in the table and the module counts in the 'Modules' column.

Name	Path	Deployment Order	Modules
opencrx-core-CRX-App	C:\p\gm\bea\user_projects\domains\crx\myserver\upload\opencrx-core-CRX-App.ear	100	5
opencrx-core-CRX-war	C:\p\gm\bea\user_projects\domains\crx\myserver\upload\opencrx-core-CRX-war.ear	100	1

## Chapter 9. Next Steps

Before you proceed to the *openCRX QuickStart guide* make sure that you have deployed and started all applications.

The application is initialized the first time a user calls the login page. If the startup fails you should consult the following log files:

- **WebLogic console and access logs.** Located in *bea\user\_projects\domains\mydomain\myserver* and contains the console and the access log.
- **openCRX opencrx-server1...log.** Located in *bea\user\_projects\domains\mydomain\log* and contains the *openCRX* application log files.

# 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>)