

openCRX Installation Guide for BEA WebLogic Server 9.2

Version 1.10.0



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 2007 © CRIXP Corp. All rights reserved.



Table of Contents

1	About this Book.....	4
1.1	Who this book is for.....	4
1.2	What do you need to understand this book.....	4
1.3	Tips, Warnings, etc.....	4
2	Prerequisites.....	5
3	Installing openCRX for Bea WebLogic Server 9.2.....	6
4	Installing Libraries.....	7
5	Configuring the Datasource.....	8
6	Configuring Security.....	12
7	Deploying openCRX.....	16
8	Final Steps.....	22

List of Figures

Figure 1:	Bea WebLogic Server 9.2 Console.....	6
Figure 2:	Create a new JDBC data source.....	8
Figure 3:	Specify the database and login.....	9
Figure 4:	Complete creation of JDBC data source.....	10
Figure 5:	Deploy the datasource to a target server.....	11
Figure 6:	Enable WebLogic security.....	12
Figure 7:	Create global roles.....	13
Figure 8:	Add conditions.....	14
Figure 9:	Add groups to users.....	15
Figure 10:	Deploy the opencrx-core-CRX-App EAR.....	16
Figure 11:	Select Deploy EJBs.....	17
Figure 12:	Install deployment as application.....	17
Figure 13:	Deployment optional settings.....	18
Figure 14:	Verify the deployment settings.....	19
Figure 15:	Deployment of opencrx-core-CRX-Web.ear.....	20
Figure 16:	Activate changes.....	20
Figure 17:	Deploy the Web EAR and check show me all installation options...	21
Figure 18:	Enterprise applications in state Active.....	21

List of Listings

Listing 1: startWebLogic.cmd with openCRX-specific properties..... 7

1 About this Book

This book describes the installation of *openCRX* for *BEA WebLogic Server 9.2*.

1.1 Who this book is for

The intended audience are *openCRX* and application server system administrators.

1.2 What do you need to understand this book

This book describes the installation of *openCRX* for *BEA WebLogic Server 9.2*. The book assumes that you are familiar with *BEA WebLogic Server 9.2* administration and deployment concepts.

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. - it goes without saying that we advise to follow our guides meticulously

meticulous \muh-TIK-yuh-luhs\, *adjective*:
Extremely or excessively careful about details.



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 you must download and install the following software:

- Install *BEA WebLogic Server 9.2*.
- Download **openMDX** from *here*
(http://sourceforge.net/project/showfiles.php?group_id=75132).
- Download **openCRX** from *here*
(http://sourceforge.net/project/showfiles.php?group_id=95219).



BEA WebLogic Server 9.2 comes with *JRE-1.5*. However, download the *JRE-1.4* version of *openCRX* and *openMDX* (e.g. *opencrx-1.10.0-core.CRX.jre-1.4.zip*).



Before you can install *openCRX* for *WebLogic* you must install the database as described in the *openCRX* database installation guides. If you have successfully installed the *openCRX* database you are ready to continue with the *BEA WebLogic Server 9.2* setup.

3 Installing openCRX for *Bea WebLogic Server 9.2*

In a first step install *WebLogic*. You should be able to start and stop *WebLogic* and launch the *WebLogic Server Console* as shown below:

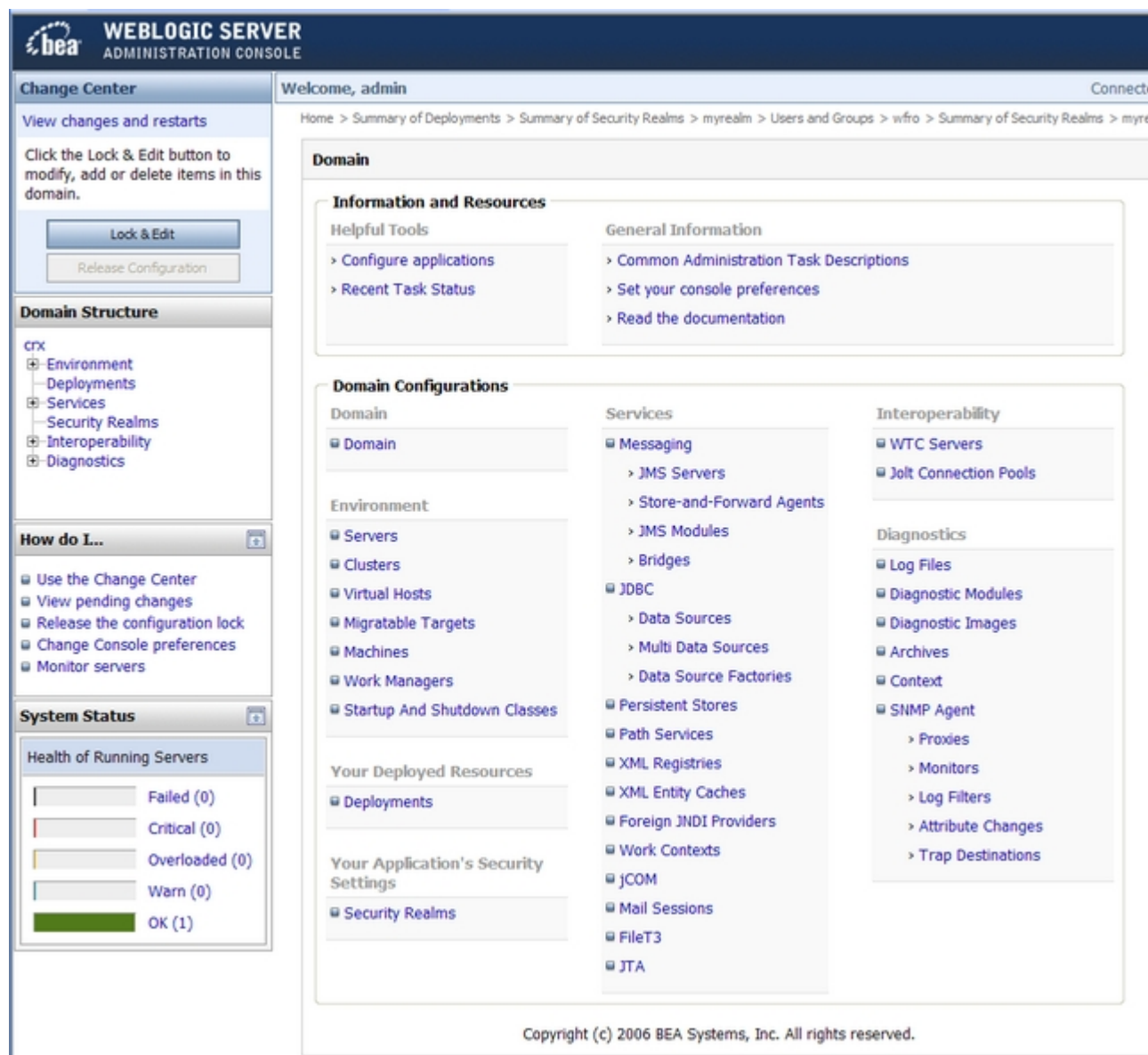


Figure 1: *Bea WebLogic Server 9.2 Console*.

The *openCRX* installation requires the following steps:

- Configure properties for the *JVM* 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*.

4 Installing Libraries

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 *PostgreSQL* the file is *postgresql-8.1-407.jdbc3.jar*.

Then you must add *openCRX*-specific options to the *WebLogic* startup script. Modify *startWebLogic.cmd* as shown in Listing 1. Restart *WebLogic* in order to activate the settings.

Listing 1: startWebLogic.cmd with openCRX-specific properties.

```
@ECHO OFF
@REM WARNING: This file is created by the Configuration Wizard.
@REM Any changes to this script may be lost when adding extensions to this configuration.
SETLOCAL
set DOMAIN_HOME=C:\pgm\bea920\user_projects\domains\crx
set EXT_POST_CLASSPATH=%DOMAIN_HOME%\lib\openmdx-kernel.jar
set EXT_POST_CLASSPATH=%EXT_POST_CLASSPATH%;%DOMAIN_HOME%\lib\openmdx-weblogic.security.jar
set EXT_POST_CLASSPATH=%EXT_POST_CLASSPATH%;%DOMAIN_HOME%\lib\postgresql-8.1-407.jdbc3.jar
set JAVA_OPTIONS=
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 MEM_ARGS=-Xms96m -Xmx1024m
call "%DOMAIN_HOME%\bin\startWebLogic.cmd" %*
ENDLOCAL
```

5 Configuring the Datasource

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

Navigate to *Services > JDBC > Data Sources* and click on *New*. Fill the fields as follows:

- **Name.** *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.
- **Database Type / Driver.** Select the database type and driver from the *Database Driver* drop down, e.g. *PostgreSQL / PostgreSQL's Driver (Type 4)*. Then click *Next*.

Welcome, admin Connected to: crx

Home > Summary of Deployments > Summary of Security Realms > myrealm > Users and Groups > wfro > Summary of Security Realms > myrealm > Summ

Create a New JDBC Data Source

Back Next Finish Cancel

JDBC Data Source Properties

The following properties will be used to identify your new JDBC data source.

What would you like to name your new JDBC data source?

Name:

What JNDI name would you like to assign to your new JDBC Data Source?

JNDI Name:

What database type would you like to select?

Database Type:

What database driver would you like to use to create database connections?

Database Driver:

Back Next Finish Cancel

Figure 2: Create a new JDBC data source.

On the next screen you must configure the following properties as shown below:

- **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 *PostgreSQL* this is typically 5432, 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.

Welcome, admin Connected to: crx [Home](#) [Log Out](#)

Home > Summary of Deployments > Summary of Security Realms > myrealm > Users and Groups > wfro > Summary of Deployments > crx > **Summary of JDBC Data Sources**

Create a New JDBC Data Source

[Back](#) [Next](#) [Finish](#) [Cancel](#)

Connection Properties
Define Connection Properties.

What is the name of database you would like to connect to?

Database Name:

What is the name or IP address of the database server?

Host Name:

What is the port on the database server used to connect to the database?

Port:

What database account user name do you want to use to create database connections?

Database User Name:

What is the database account password to use to create database connections?

Password:

Confirm Password:

[Back](#) [Next](#) [Finish](#) [Cancel](#)

Figure 3: Specify the database and login.

Then click *Next*. Verify the database connection properties on the next page as shown below and then click *Test Driver Configuration*.

The screenshot shows a web browser window with the following elements:

- Page Header:** "Welcome, admin" on the left, and "Connected to: crx" followed by navigation links "Home", "Log Out", "Preferences", "Help", and "AskBE" on the right.
- Breadcrumb:** "Home > Summary of Deployments > Summary of Security Realms > myrealm > Users and Groups > wfro > Summary of Security Realms > myrealm > Summary of Deployments > crx > Summary of JDBC Data Sources".
- Section Header:** "Create a New JDBC Data Source".
- Navigation:** A row of buttons: "Test Configuration", "Back", "Next", "Finish", and "Cancel".
- Section Header:** "Test Database Connection".
- Text:** "Test the database availability and the connection properties you provided."
- Question:** "What is the full package name of JDBC driver class used to create database connections in the connection pool? (Note that this driver class must be in the classpath of any server to which it is deployed.)"
- Field:** "Driver Class Name:" with the value "org.postgresql.Driver".
- Question:** "What is the URL of the database to connect to? The format of the URL varies by JDBC driver."
- Field:** "URL:" with the value "jdbc:postgresql://localhost".
- Question:** "What database account user name do you want to use to create database connections?"
- Field:** "Database User Name:" with the value "postgres".
- Question:** "What is the database account password to use to create database connections?"
- Field:** "Password:" with a masked password of 12 dots.
- Field:** "Confirm Password:" with a masked password of 12 dots.
- Question:** "What are the properties to pass to the JDBC driver when creating database connections?"
- Field:** "Properties:" with a text area containing "user=postgres".
- Question:** "What table name or SQL statement would you like to use to test database connections?"
- Field:** "Test Table Name:" with a text area containing "SQL SELECT 1".
- Navigation:** A second row of buttons: "Test Configuration", "Back", "Next", "Finish", and "Cancel".

Figure 4: Complete creation of JDBC data source.

If everything works fine you should see the *Connection successful* message.

Click *Next*. 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 below:

The screenshot shows a web browser interface for configuring a JDBC data source. At the top, a navigation bar includes 'Welcome, admin', 'Connected to: crx', and links for 'Home' and 'Log Out'. Below this is a breadcrumb trail: 'Home > Summary of Deployments > Summary of Security Realms > myrealm > Users and Groups > wfro > Summary of Security Deployments > crx > Summary of JDBC Data Sources'. The main content area is titled 'Create a New JDBC Data Source' and contains a sequence of buttons: 'Back', 'Next', 'Finish', and 'Cancel'. The 'Next' button is highlighted, indicating the current step. Below the buttons is the 'Select Targets' section, which includes a descriptive paragraph: 'You can select one or more targets to deploy your new JDBC data source. If you don't select a target created but not deployed. You will need to deploy the data source at a later time.' Underneath this text is a table with the heading 'Servers'. The table contains one entry, 'AdminServer', which has a checked checkbox to its left. At the bottom of the form, there is another set of 'Back', 'Next', 'Finish', and 'Cancel' buttons.

Figure 5: Deploy the datasource to a target server.

6 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:

- Bring up the detail page of the security realm *myrealm* by clicking *Security Realms > myrealm*.
- Then click *Users and Groups > Groups > New*. You must create the following groups: *OpenCrxUser*, *OpenCrxAdministrator*, *OpenCrxRoot* as shown below:

Welcome, admin Connected to: crx Home Log Out Preferences Help AskBEA

Home > Users and Groups > myrealm > Realm Roles > Realm Policies > Realm Roles > Users and Groups > myrealm > Summary of JDBC Data Sources > Summary of Security Realms > myrealm

Settings for myrealm

Configuration Users and Groups Roles and Policies Credential Mappings Providers Migration

Users Groups

This page displays information about each group that has been configured in this security realm.

[Customize this table](#)

Groups

New Delete Showing 1 - 8 of 8 Previous | Next

<input type="checkbox"/>	Name ↕	Description	Provider
<input type="checkbox"/>	Administrators	Administrators can view and modify all resource attributes and start and stop servers.	DefaultAuthenticator
<input type="checkbox"/>	AppTesters	AppTesters group.	DefaultAuthenticator
<input type="checkbox"/>	Deployers	Deployers can view all resource attributes and deploy applications.	DefaultAuthenticator
<input type="checkbox"/>	Monitors	Monitors can view and modify all resource attributes and perform operations not restricted by roles.	DefaultAuthenticator
<input type="checkbox"/>	OpenCrxAdministrator		DefaultAuthenticator
<input type="checkbox"/>	OpenCrxRoot		DefaultAuthenticator
<input type="checkbox"/>	OpenCrxUser		DefaultAuthenticator
<input type="checkbox"/>	Operators	Operators can view and modify all resource attributes and perform server lifecycle operations.	DefaultAuthenticator

New Delete Showing 1 - 8 of 8 Previous | Next

Figure 6: Enable WebLogic security.

Then select *Roles and Policies > Realm Roles > Global Roles*. Create the roles *OpenCrxRoot*, *OpenCrxAdministrator* and *OpenCrxUser* as shown below:

The screenshot shows the 'Roles and Policies' configuration page for a security realm. The page includes a navigation breadcrumb, a 'Settings for myrealm' header, and tabs for 'Configuration', 'Users and Groups', 'Roles and Policies', 'Credential Mappings', 'Providers', and 'Migration'. Under 'Roles and Policies', there are sub-tabs for 'Realm Roles' and 'Realm Policies'. A text block explains that the table below is used to view, add, modify, or remove global or scoped security roles. Notes indicate that the table does not list scoped roles for JNDI or WorkContext resources and that imported roles for EJBs or Web applications must be activated. The 'Roles' section features an 'Edit Role' button and a table with 7 columns: Name, Resource Type, and Role Policy. The table lists various roles, with 'OpenCrxAdministrator', 'OpenCrxRoot', and 'OpenCrxUser' highlighted in blue. The table also shows a 'Showing 1 - 7 of 7' indicator and 'Previous | Next' navigation links.

Name	Resource Type	Role Policy
Deployments		
Domain		
Global Roles		
Roles		
Admin	Global Role	View Role Conditions
Anonymous	Global Role	View Role Conditions
AppTester	Global Role	View Role Conditions
Deployer	Global Role	View Role Conditions
Monitor	Global Role	View Role Conditions
OpenCrxAdministrator	Global Role	View Role Conditions
OpenCrxRoot	Global Role	View Role Conditions
OpenCrxUser	Global Role	View Role Conditions
Operator	Global Role	View Role Conditions
JCOM		
JDBC		
JMS		
Servers		

Figure 7: Create global roles.

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

- Select the global role *OpenCrxUser* and click *Add Conditions*. Select the predicate *Group* in the drop-down and then click *Next*. Enter *OpenCrxUser* in the field *Group Argument Name* and then click *Add*.
- Select the global role *OpenCrxAdministrator* and click *Add Conditions*. Select the predicate *Group* in the drop-down and then click *Next*. Enter *OpenCrxAdministrator* in the field *Group Argument Name* and then click *Add*.
- Select the global role *OpenCrxRoot* and click *Add Conditions*. Select the predicate *Group* in the drop-down and then click *Next*. Enter *OpenCrxRoot* in the field *Group Argument Name* and then click *Add*.

The global role *OpenCrxAdministrator* should then look as shown below:

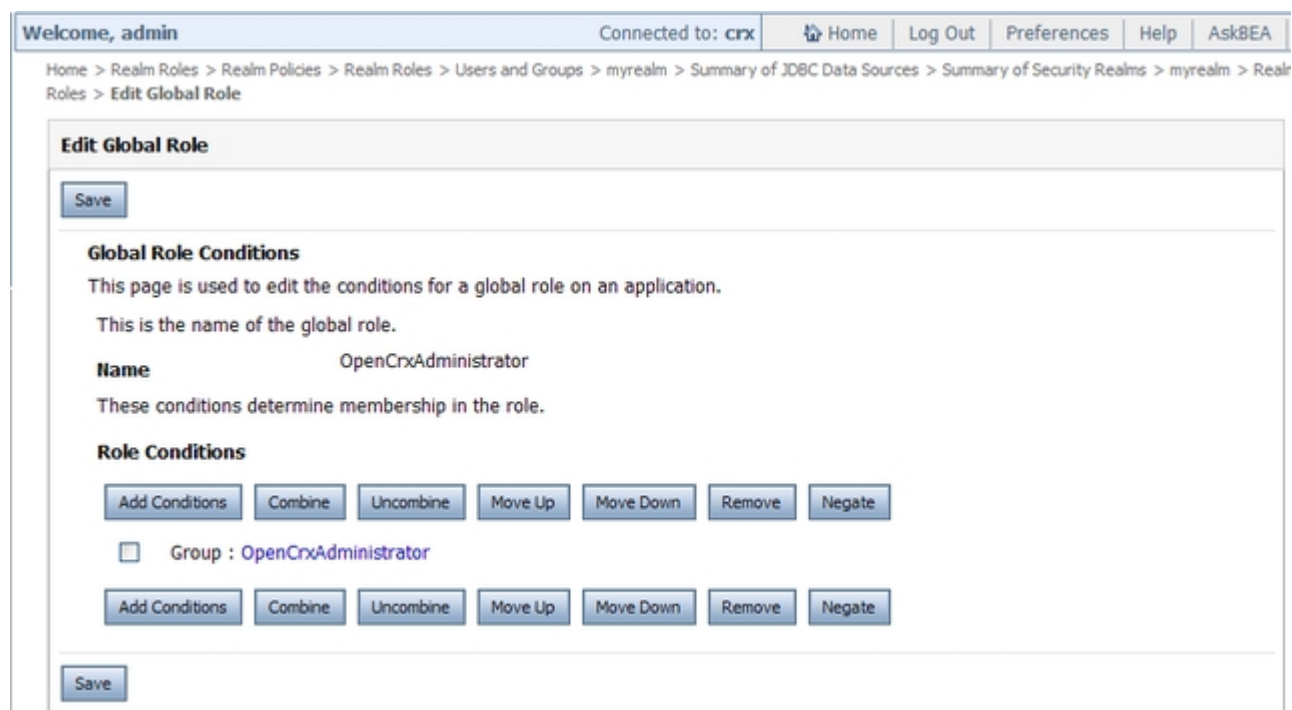


Figure 8: Add conditions.

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

You are now ready to create users and make them member of the groups *OpenCrxRoot*, *OpenCrxAdministrator* or *OpenCrxUser*. The servlets in EAR *opencrx-core-CRX-Web.ear* require the users to be member of the groups *OpenCrxUser*, *OpenCrxAdministrator* or *OpenCrxRoot*. Standard users are member of the group *OpenCrxUser*. Segment administrators are member of the group *OpenCrxAdministrator*. This is shown in the figure below.

As a start 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*.
- **guest:** guest user. Make it member of *OpenCrxUser*.

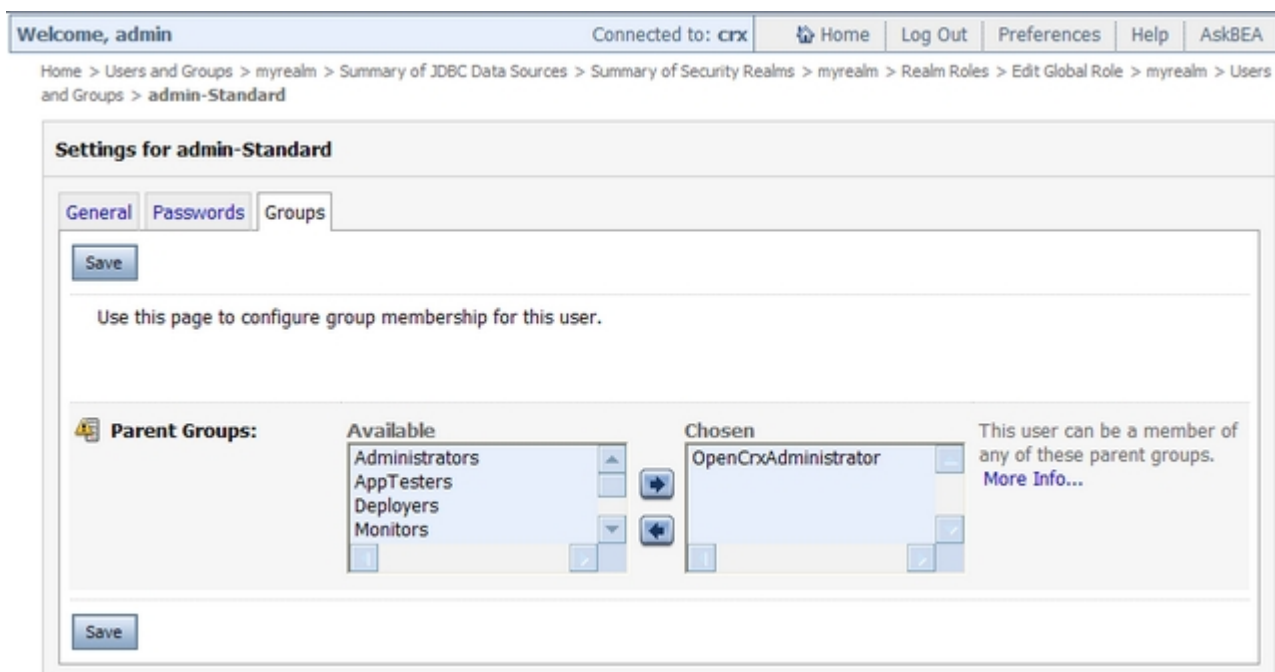


Figure 9: Add groups to users.

7 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 *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 below:

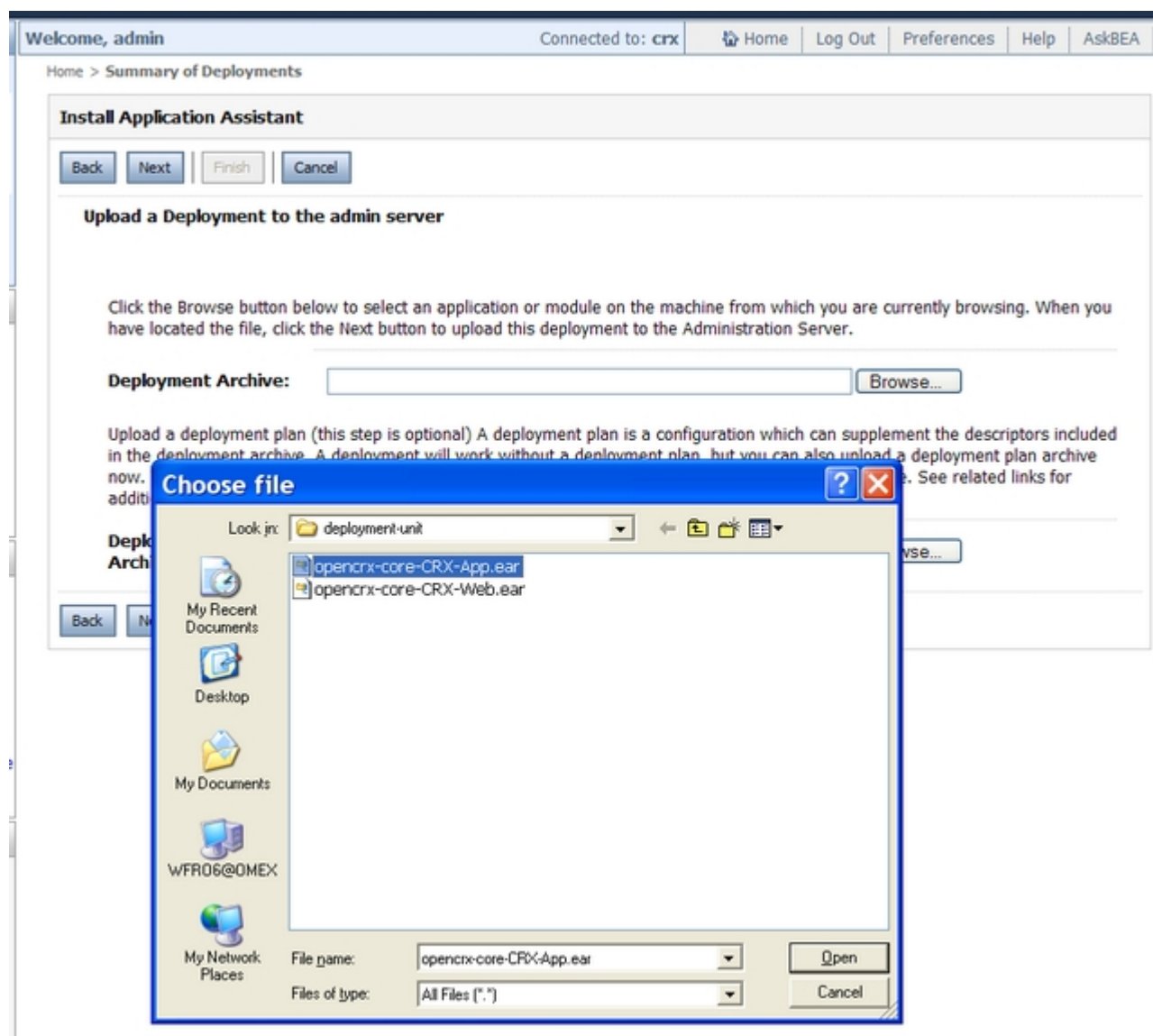


Figure 10: Deploy the *opencrx-core-CRX-App* EAR.

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

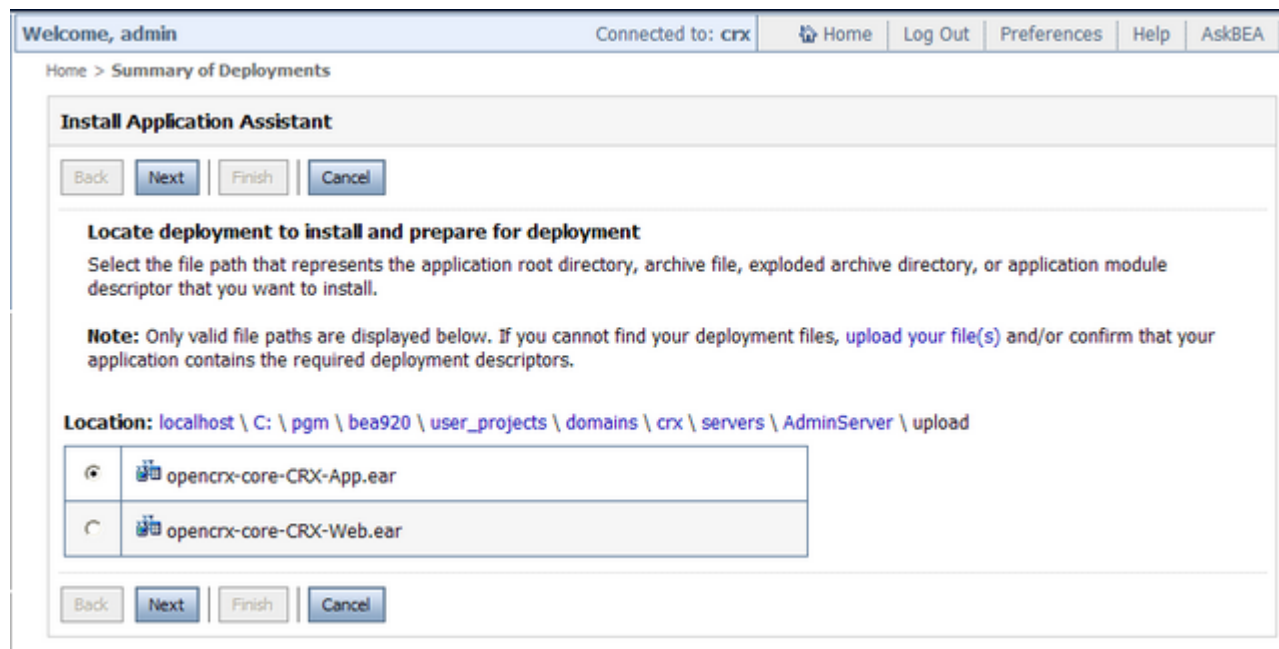


Figure 11: Select Deploy EJBs.

Then click *Next*. On the next screen you must select whether to install the archive as library or application. Check *Install this deployment as an application* as shown below:

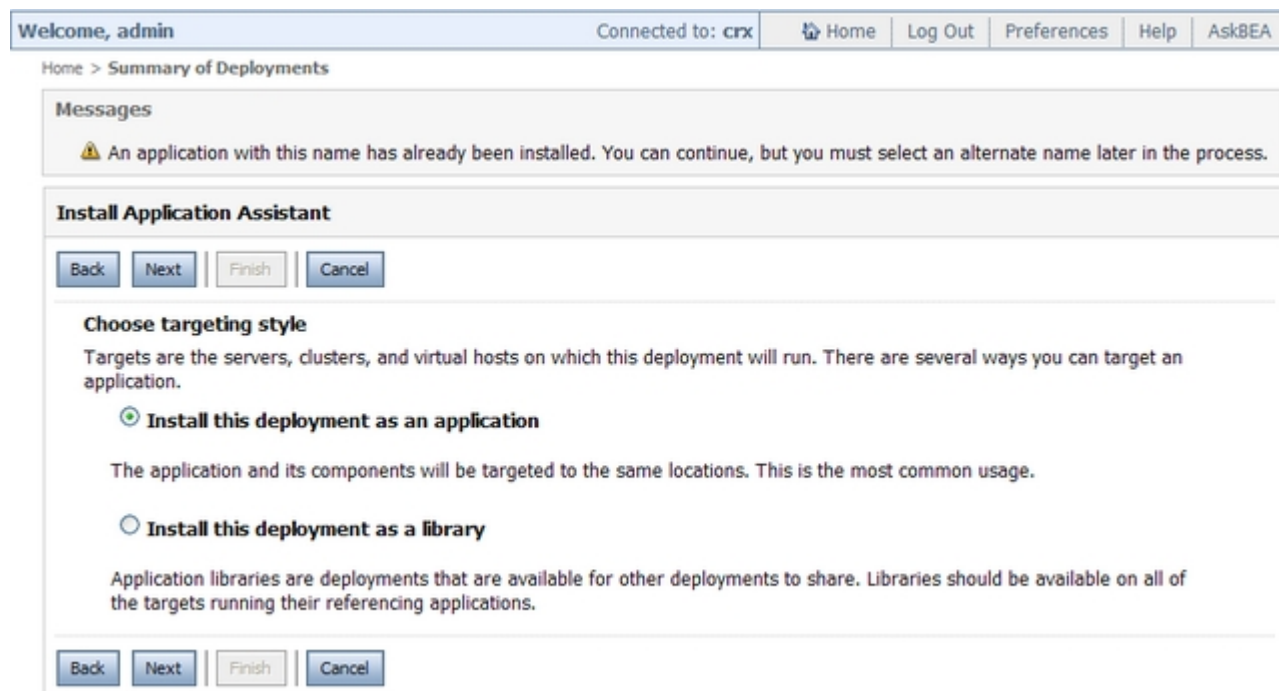


Figure 12: Install deployment as application.

Click *Next*. On the next screen specify optional settings. Make sure that the security model *DD only* is selected as shown below. Leave the application name unchanged.

The screenshot shows the 'Install Application Assistant' dialog box with the following sections and settings:

- Optional Settings:** You can modify these settings or accept the defaults.
- General:**
 - What do you want to name this deployment?
Name:
 - Specification Version:** 1.9
 - Implementation Version:** 1.9.12+-20070214T093834Z
- Security:**
 - What security model do you want to use with this application?
 - DD Only: Use only roles and policies that are defined in the deployment descriptors.**
 - Custom Roles: Use roles that are defined in the Administration Console; use policies that are defined in the deployment descriptor.
 - Custom Roles and Policies: Use only roles and policies that are defined in the Administration Console.
 - Advanced: Use a custom model that you have configured on the realm's configuration page.
- Source accessibility:**
 - How should the source files be made accessible?
 - Use the defaults defined by the deployment's targets**
Recommended selection.
 - Copy this application onto every target for me
During deployment, the files will be copied automatically to the managed servers to which the application is targeted.
 - I will make the deployment accessible from the following location
Location:
 - Provide the location from where all targets will access this application's files. This is often a shared directory. You must ensure the application files exist in this location and that each target can reach the location.

Figure 13: Deployment optional settings.

Click *Next*.

On the next screen specify the deployment targets and the application name. Leave the default values unchanged and then click *Deploy* as shown below:

The screenshot shows the 'Install Application Assistant' web interface. At the top, there is a navigation bar with 'Welcome, admin', 'Connected to: crx', and links for 'Home', 'Log Out', 'Preferences', 'Help', and 'AskBEA'. Below this is a breadcrumb 'Home > Summary of Deployments'. The main content area is titled 'Install Application Assistant' and contains several sections:

- Buttons:** 'Back', 'Next', 'Finish', and 'Cancel'.
- Review your choices and click Finish:** A message stating 'Click Finish to complete the deployment. This may take a few moments to complete.'
- Additional configuration:** A section asking 'In order to work successfully, this application may require additional configuration. Do you want to review this application's configuration after completing this assistant?' with two radio button options:
 - Yes, take me to the deployment's configuration screen.
 - No, I will review the configuration later.
- Summary:** A table-like summary of deployment settings:
 - Deployment:** C:\pgrm\bea920\user_projects\domains\crx\servers\AdminServer\upload\opencrx-core-CRX-App.ear
 - Name:** opencrx-core-CRX-App
 - Staging mode:** Use the defaults defined by the chosen targets
 - Security Model:** DDOnly: Use only roles and policies that are defined in the deployment descriptors.
- Target Summary:** A table with two columns: 'Modules' and 'Targets'.

Modules	Targets
opencrx-core-CRX-App	AdminServer
- Buttons:** 'Back', 'Next', 'Finish', and 'Cancel'.

Figure 14: Verify the deployment settings.

Repeat the deployment step for the archive *opencrx-core-CRX-Web.ear*. The summary screen should look as shown below:

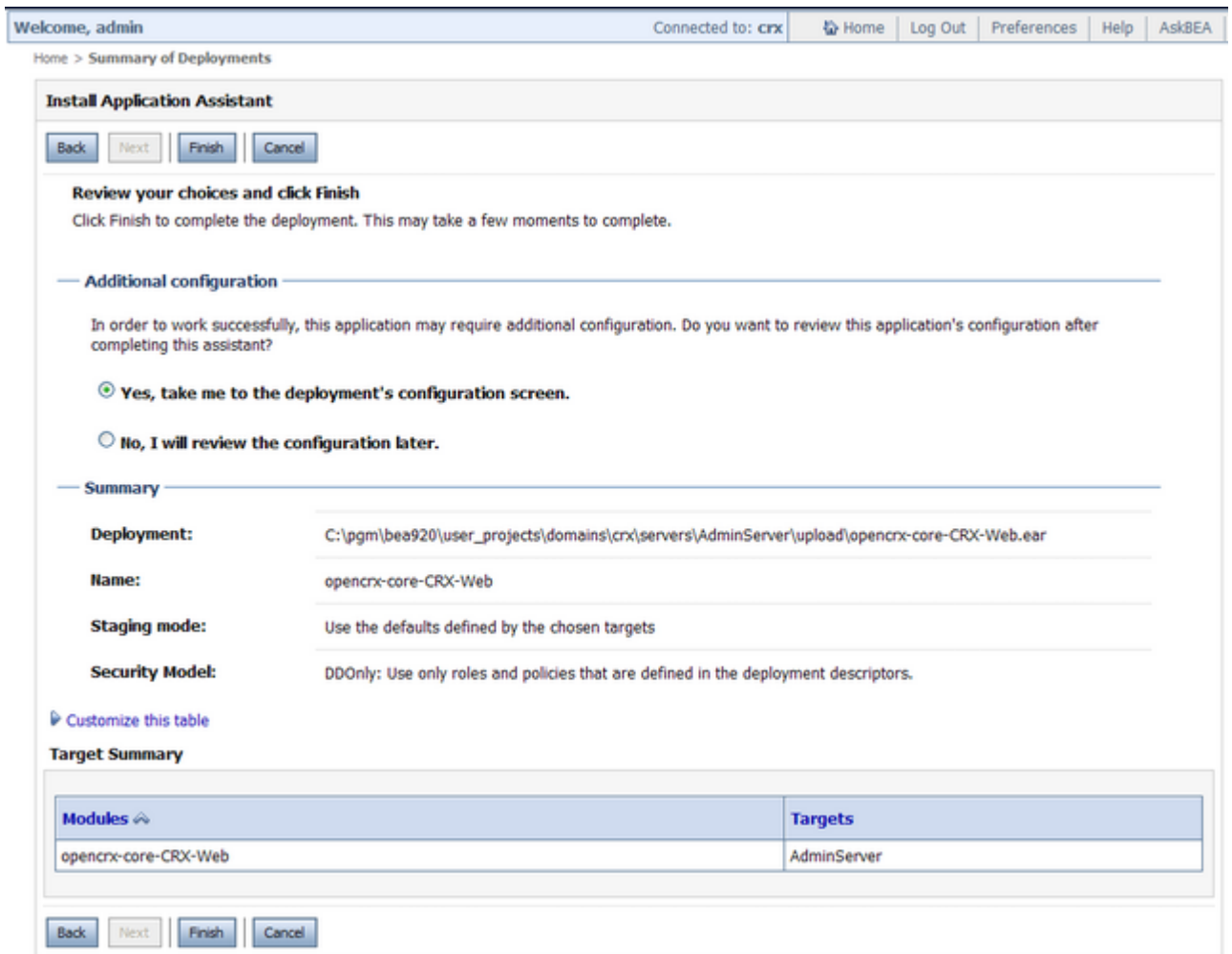


Figure 15: Deployment of *opencrx-core-CRX-Web.ear*

Do not forget to activate the changes by clicking on *Activate Changes*:

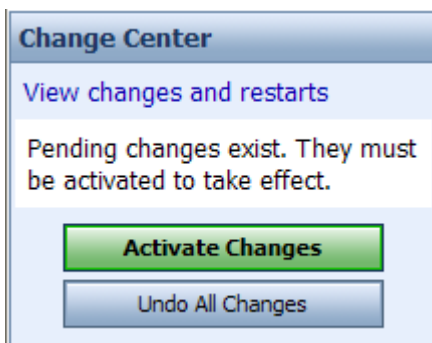


Figure 16: Activate changes.

Finally, the the deployed applications should look as shown below:

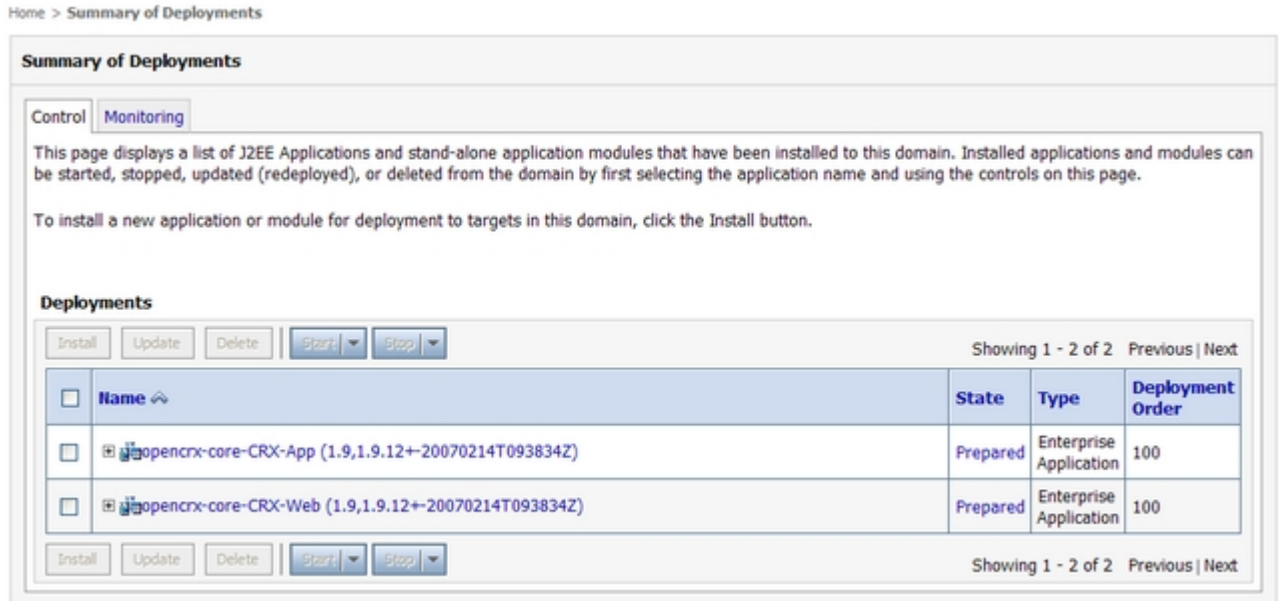


Figure 17: Deploy the Web EAR and check show me all installation options.

Before you can use *openCRX* you must start the applications *opencrx-core-CRX-App* and *opencrx-core-CRX-Web*. Check the corresponding boxes and then select *Start > Servicing all requests*. Make sure that the state changes to *Active* as shown below:

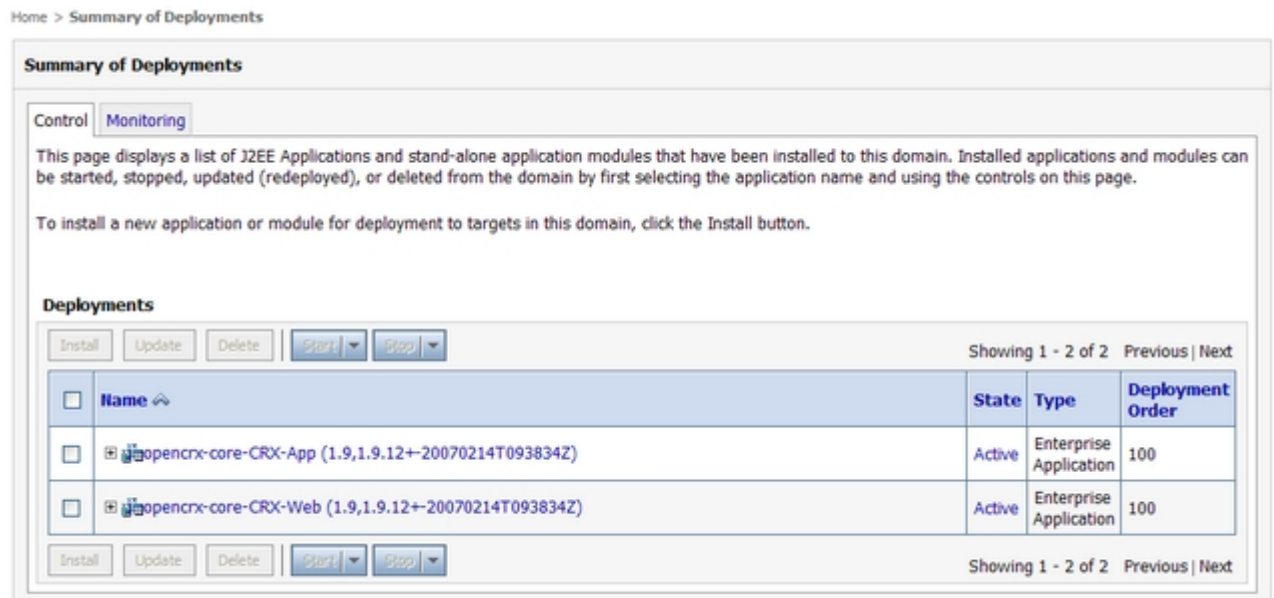


Figure 18: Enterprise applications in state Active.

8 Final 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 consult the following log files:

- **WebLogic console and access logs.** Located in *bea\user_projects\domains\mydomain\servers\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.