BEA WebLogic Server™

MedRec Clustering Tutorial

Release 8.1
Document Date: February 2003
Revised: July 18, 2003

Copyright

Restricted Rights Legend

Trademarks or Service Marks

MedRec Tutorials

Part Number	Date	Software Version
N/A	March 28, 2003	BEA WebLogic Server Version 8.1

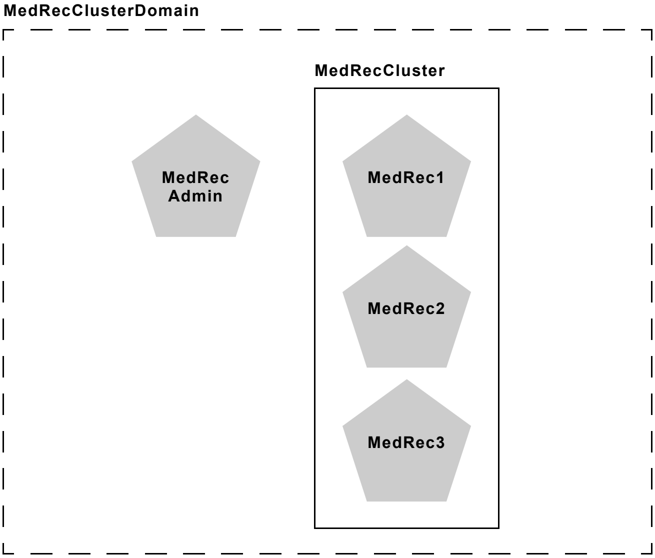
Creating and Deploying to WebLogic Server Clusters

Overview of the Clustering Tutorials

These tutorials guide you through the process of creating and configuring a new cluster using the WebLogic Configuration Wizard and Administration Console. They also explain how to deploy the Avitek Medical Records sample application on the new cluster.

The cluster you will create in this tutorial contains an Administration Server and three Managed Servers as shown in the figure below. In these tutorials, you create the entire cluster on one machine. However, production clusters generally include multiple machines and/or specialized hardware.

Figure 1: MedRecClusterDomain



You will also need to run the PointBase database on the cluster machine. If your machine does not have enough resources to host all servers, you can create only two servers, Medrec1 and MedRec2, in the cluster.

Creating and Deploying to WebLogic Server Clusters

Tutorial 1: Creating a WebLogic Server Cluster

In this tutorial you use the WebLogic Server Configuration Wizard to create the domain, servers, and cluster in which you will deploy and run the MedRec applications. The new domain, medrecdomain, contains a single Administration Server and a cluster of three Managed Servers.

The tutorial includes the following sections:

- Prerequisites
- Procedure
- Best Practices
- The Big Picture
- Related Reading

Prerequisites

Before starting this tutorial, make sure WebLogic Server and the server samples are installed on your computer.

Procedure

To create the domains server instances, and clusters:

1. Start the Configuration Wizard on your system by selecting the Start Menu option:

   Start→Programs→BEA WebLogic Platform 8.1→Configuration Wizard

2. On the initial screen, select Create a new WebLogic configuration.
3. On the Select a Configuration Template screen, select the Basic WebLogic Server Domain template.

   Note: Do not select the Avitek Medical Records Sample Domain template, as the tutorial does not make use of its functionality.

4. On the Choose Express or Custom Configuration screen, select Custom and click Next..
5. On the Configure the Administration Server screen:
   - Enter medrecadmin in the Name field.
   - Enter 127.0.0.1 in the ListenAddress field.
   - Enter 7001 in the Listen Port field.
   - Click Next..
6. On the Managed Servers, Clusters, and Machines Options screen, select Yes and click Next to continue..
7. On the Configure Managed Servers screen, click Add and:
   - Enter MedRec1 in the Name field.
   - Enter 127.0.0.1 in the ListenAddress field.
   - Enter 8001 in the ListenPort field.
   - Repeat this step to add two additional servers with the following field entries:

Name	ListenAddress	ListenPort
MedRec2	127.0.0.1	8011

Name	ListenAddress	ListenPort
MedRec3	127.0.0.1	8021

- Click Next to continue.

8. On the Configure Clusters screen, click Add and:
   - Enter MedRecCluster in the Name field.
   - Enter 237.0.0.101 in the MulticastAddress field
   - Enter 8050 in the Multicast Port field.
   - Click Next to continue.
9. In the Assign Servers to Clusters screen:
   - Select all three servers in the Source list.
   - Click the arrow button to place the servers in the Target list.
   - Click Next to continue.
10. On the Configure Machines screen, click Next without adding any machines.
11. On the Database (JDBC) Options screen, click Next.
12. On the Messaging (JMS) Options screen, click Next.
13. On the Configure Administrative Username and Password screen:
    - Enter weblogic in the Name field.
    - Enter weblogic in the Password field.
    - Enter weblogic in the Confirm User Password field.
    - Click Next to continue.
14. On the Configure Windows Options screen:
    - Select No in the Create Start Menu section.
    - Select No in the Install Administration Server as a Windows Service section.
    - Click Next to continue.
15. On the Configure Server Start Mode and Java SDK screen:

- Select Development Mode in the WebLogic Configuration Startup Mode section.
- Click Next to continue.

16. On the Create WebLogic Configuration screen:
    - Enter MedRecClusterDomain in the Configuration Name field.
    - Click the Create button to create the new domain.
    - Click Done to exit the Configuration Wizard.

    The Configuration Wizard creates the new domain in `c:\bea\user_projects\domains\MedRecClusterDomain`.

Best Practices

- When creating the clustered WebLogic Server instances on a single machine, specify an unused localhost port such as 127.0.0.1 for the Listen Address. If you leave the Listen Address field blank, the server binds to all available IP addresses on the machine. If you specify "localhost" as the Listen Address, the server binds to all localhost addresses (127.0.0.x).
- Always specify an unused Multicast address and port number for production clusters. This ensures that only the cluster's messages are broadcast on the address/port combination.

The Big Picture

This tutorial is the basis for setting up your cluster environment. Before you can deploy applications to a server, you must first configure the domains and servers to which you want to deploy the applications.

The servers that you created have a minimal configuration—you specified only the server names and listen address, and their membership in the cluster. In the next tutorial, you will use the Administration Console to create and assign the server resources necessary to deploy the MedRec application suite on the cluster.

Related Reading

- *Creating and Configuring Domains Using the Configuration Wizard* in *Configuring and Managing WebLogic Server*
- Starting and Stopping WebLogic Servers: Quick Reference in the *Configuring and Managing WebLogic Server*

1 Creating and Deploying to WebLogic Server Clusters

Tutorial 2: Starting the Cluster

This tutorial describes how to start the new cluster so that you can configure MedRec resources and deploy the MedRec applications.

The tutorial includes the following sections:

- Prerequisites
- Procedure
- Best Practices
- The Big Picture
- Related Reading

Prerequisites

Before starting this tutorial, create the MedRecCluster domain. See Tutorial 1: Creating a WebLogic Server Cluster.

Procedure

Follow these steps to start the MedRec cluster:

- Step 1: Start PointBase.
- Step 2: Start the Administration Server for the cluster domain.
- Step 3: Edit the Managed Server startup scripts.
- Step 4: Start the Managed Servers in the cluster.
- Step 5: Invoke the Administration Console for MedRecServer in your browser.

Step 1: Start PointBase.

Before configuring resources in the cluster, start the PointBase server in order to test the JDBC configuration in the steps that follow.

1. Open a new command window by selecting Start—Run... and entering cmd.exe in the resulting dialog box.
2. Move to the WebLogic Server 8.1 `tools` subdirectory:

   ```
   cd c:\bea\weblogic81\common\eval\pointbase\tools
   ```

3. Start PointBase by executing its start script:

   ```
   startPointBase.cmd
   ```

   The server starts and displays the message:

   ```
   Server started, listening on port 9092, display level: 3 ...
   ```

Step 2: Start the Administration Server for the cluster domain.

You will configure cluster resources using the Administration Console for the domain you created in the previous tutorial. To use the Administration Console, first start the Administration Server for MedRecClusterDomain:

1. Open a new command window by selecting Start—Run... and entering cmd.exe in the resulting dialog box.
2. Move to the new MedRecClusterDomain directory you created in the previous tutorial:

```
cd c:\bea\user_projects\domains\MedRecClusterDomain
```

3. Open three additional command windows by entering the commands:

   ```
   start cmd
   ```

   ```
   start cmd
   ```

   ```
   start cmd
   ```

   You will use these additional command windows to start the Managed Servers in in Step 4: Start the Managed Servers in the cluster.

4. Start the Administration Server by executing its start script:

   ```
   startWebLogic.cmd
   ```

   The server displays boot messages and finishes with the message:

   ```
   <Feb 19, 2003 2:10:08 PM PST> <Notice> <WebLogicServer>
   <BEA-000360> <Server started in RUNNING mode>
   ```

Step 3: Edit the Managed Server startup scripts.

The MedRec application suite uses log4j for logging application messages. You must copy the log4j properties file from the installed MedRec domain and identify it using a startup option in the Managed Server startup scripts. For Web Services, you must also identify the `.wsdl` and the incoming directory for XML files. To complete these steps:

1. Copy the log4j properties file from the pre-configured MedRec domain to the new domain you just created:

   ```
   cp c:\bea\weblogic81\samples\domains\medrec\log4j.properties
   c:\bea\user_projects\domains\MedRecClusterDomain
   ```

2. Open the `startManagedWebLogic.cmd` script for your new domain:

   ```
   notepad
   c:\bea\user_projects\domains\MedRecClusterDomain\startManagedWe
   bLogic.cmd
   ```

3. Find the following comment lines in the `startManagedWebLogic.cmd` script:

   ```
   @rem Set JAVA_OPTIONS to the java flags you want to pass to the
   vm. i.e.:
   ```

   ```
   @rem set JAVA_OPTIONS=-Dweblogic.attribute=value
   -Djava.attribute=value
   ```

4. Add the following line immediately after the above comments:

   ```
   set JAVA_OPTIONS=-Dlog4j.config=log4j.properties
   -Dcom.bea.medrec.xml.incoming=incoming
   -Dphys.app.wsdl.url=http://localhost:8001/ws_medrec/MedRecWebSe
   rvices?WSDL
   ```

5. Save the file and exit your text editor.

Step 4: Start the Managed Servers in the cluster.

1. In one of the command windows you opened in Step 2: Start the Administration Server for the cluster domain., enter the following command to start the MedRec1 server:

   ```
   startmanagedweblogic MedRec1 http://127.0.0.1:7001
   ```

2. Repeat Step 1 with the remaining command windows to start the other servers in the cluster. Use the commands:

   ```
   startmanagedweblogic MedRec2 http://127.0.0.1:7001
   ```

   ```
   startmanagedweblogic MedRec3 http://127.0.0.1:7001
   ```

Step 5: Invoke the Administration Console for MedRecServer in your browser.

Use the Administration Console to verify that Managed Server instances in the cluster have booted successfully.

1. Open a browser window and enter the following URL:

   ```
   http://127.0.0.1:7001/console
   ```

2. Enter `weblogic` as the username and password, then click **Sign In**.
3. Expand the Clusters node in the left pane and select MedRecCluster.
4. Click the Monitoring tab in the right pane, and monitor the startup process for the Managed Servers. Continue with the next tutorial when all servers in the cluster show the “RUNNING” state. (Use the refresh button in the right pane to ensure that the contents of the monitoring page update at regular intervals.)

Best Practices

The Configuration Wizard installs the `startManagedWebLogic.cmd` script to help you start Managed Servers in a multi-server domain. When using this script, you need only specify the name of the server to start, and the URL to contact the Administration Server for the domain.

The Big Picture

The cluster you have created and started represents a typical cluster used on a development or multihomed machine. The configuration and root directories for all servers reside on the same physical machine, and each server listens on its own address.

In a development environment, localhost addresses can be used to provide unique listen addresses for each server (as they are in this tutorial). In a production environment, a multihomed machine would generally include multiple Network Interface Cards (NICs) or a multihomed NIC to provide the cluster addresses, as well as multiple CPUs to manage the load of the clustered servers.

Related Reading

- *PointBase Console Guide* at http://e-docs.bea.com/wls/docs81/pdf/pbconsole.pdf
- Cluster Architecture in *Using WebLogic Server Clusters*

1 Creating and Deploying to WebLogic Server Clusters

Tutorial 3: Setting Up Cluster Resources for MedRec

This tutorial describes how to set up the WebLogic Server resources required to deploy and run the MedRec applications. These resources include:

- Java Database Connectivity (JDBC) connection pools and data sources for the PointBase database management system
- Java Message Service (JMS) persistent store, JMS server, queue, and connection factory
- JavaMail mail sessions

The tutorial includes the following sections:

- Prerequisites
- Procedure
- Best Practices
- The Big Picture

- Related Reading

Prerequisites

Before starting this tutorial:

- Create the MedRecClusterDomain as described in Tutorial 1: Creating a WebLogic Server Cluster.
- Start the PointBase Database, Administration Server, and Cluster using the instructions in Tutorial 2: Starting the Cluster.
- Make sure you have the Administration Console opened in a browser window and that all servers in the cluster are running. See Tutorial 2: Starting the Cluster for information on starting the Administration Console.

Procedure

Follow these steps to configure WebLogic Server resources for the MedRec Server:

- Step 1: Create a JDBC connection pool.
- Step 2: Create a JDBC DataSource.
- Step 3: Create a JMS Connection Factory.
- Step 4: Create a JMS JDBC store.
- Step 5: Create a JMS server.
- Step 6: Create the JMS queues.
- Step 7: Add email capabilities to the MedRec application.
- Step 8: Configure the MedRec Sample Authenticator.

Step 1: Create a JDBC connection pool.

This JDBC connection pool uses a JDBC driver that does not support distributed transactions (non-XA).

1. In the left pane of the Administration Console, open Services-> JDBC.
2. Right-click Connection Pools and choose **Configure a new JDBCConnectionPool**.
3. Choose `Pointbase` for the Database Type.
4. Choose `Pointbase's Driver (Type 4) Versions:4.X` for the Database Driver.
5. Click Continue.
6. Enter:
   - `pb/MedRecPool` in the Name field.
   - `demo` in the Database Name field.
   - `127.0.0.1` in the Host Name field.
   - `9092` in the Port field.
   - `MedRec` in the Database User Name field.
   - `MedRec` in the Password field.
   - `MedRec` in the Confirm Password field.
7. Click Continue.
8. Ensure that the information to test the connection to the Pointbase database is correct, then click Test Driver Configuration.

   Note: Be sure you have started PointBase, or the test of its driver configuration will fail.
9. After verifying that the connection succeeded, the Administration Console display the Create and Deploy page. To deploy to the new cluster:
   - Select medreccluster in the list of clusters.
   - Select All servers in the cluster is selected.
10. Click Create and Deploy to deploy the new pool.
11. Verify that the pool deployed by examining its entry in the JDBC Connection Pools table. pb/MedRecPool should appear in the table, and its Deployed status should be "True."

Step 2: Create a JDBC DataSource.

Client and server-side JDBC applications obtain a DBMS connection through a DataSource. A DataSource is an interface between an application and the JDBC connection pool.

1. In the left pane of the Administration Console, open Services-> JDBC.
2. Right-click Data Sources and choose **Configure a new JDBCTXDataSource**.
3. Enter `pb/MedRecTxDataSource` in the Name field and `MedRecTxDataSource` in the JNDI Name field.
4. Ensure that Honor Global Transactions is checked.
5. Click Continue.
6. Choose `pb/MedRecPool` in the Pool Name list box.
7. Click Continue.
8. Ensure that medreccluster is checked as the cluster on which you want to deploy this data source. Also ensure that the All servers in the cluster option is selected.
9. Click Create.
10. To verify that the DataSource was deployed:
    - Open the Clusters->medreccluster node in the left pane.
    - Select one of the servers in the cluster.
    - In the right pane, select the View JNDI tree link.
    - In the resulting window, verify that MedRecTxDataSource appears in the left column. This indicates that it has been bound to the cluster JNDI tree.

Step 3: Create a JMS Connection Factory.

1. In the left pane of the Administration Console, open Services-> JMS.
2. Right-click Connection Factories and choose **Configure a new JMSConnectionFactory**.
3. Enter `jms/MedRecQueueConnectionFactory` in the Name field and `jms/MedRecQueueConnectionFactory` in the JNDI Name field.

4. Click the Create button.
5. Select medreccluster in the Targets list.
6. Ensure that the All servers in the cluster option is selected.
7. Click Apply to target the connection factory to the cluster.
8. Select the Configuration tab.
9. Select the Transactions tab.
10. Select the XA Connection Factory Enabled check box.
11. Click Apply.
12. To verify that the factory was deployed:
    - Open the Clusters->medreccluster node in the left pane.
    - Select one of the servers in the cluster.
    - In the right pane, select the View JNDI tree link.
    - In the resulting window, open the jms node in the left pane.
    - Verify that `MedRecQueueConnectionFactory` appears in the left column under the jms node. This indicates that the factory has been bound to the cluster JNDI tree.

Step 4: Create a JMS JDBC store.

A JMS JDBC store consists of a JDBC database that can be used for persistent messaging.

1. In the left pane of the Administration Console, open Services-> JMS.
2. Slect Stores in the left pane.
3. In the right pane, choose **Configure a new JMS JDBC Store**.
4. Enter `MedRecJMSJDBCStore` in the Name field.
5. Choose `pb/MedRecPool` in the Connection Pool list box.
6. Enter `MedRec` in the Prefix Name field.
7. Click Create.

Step 5: Create a JMS server.

A JMS server manages connections and message requests on behalf of JMS clients.

1. In the left pane of the Administration Console, open Services-> JMS.
2. Right-click Servers and choose **Configure a new JMSServer**.
3. Enter `MedRecJMSServer` in the Name field.
4. Choose `MedRecJMSJDBCStore` in the Store list box.
5. Click Create.
6. Choose one of the available servers in the cluster—MedRec1, MedRec2, or MedRec3—from the target list. Do not select the migratable servers listed. That is, do not select MedRec1 (migratable), MedRec2 (migratable), or MedRec3 (migratable).
7. Click Apply.
8. To verify that the server was targeted:
   - Click the Monitoring tab.
   - Select the Monitor Active JMS Servers... link.
   - Verify the MedRecJMSServer appears in the table.

Step 6: Create the JMS queues.

JMS queues are based on the point-to-point (PTP) messaging model, which enables the delivery of a message to exactly one recipient. A queue sender (producer) sends a message to a specific queue. A queue receiver (consumer) receives messages from a specific queue.

The following procedure describes how to create the three JMS queues used by message-driven beans for registering new users of the MedRec application, handling email, and uploading XML files.

1. In the left pane of the Administration Console, select Services-> JMS->Servers->MedRecJMSSserver.
2. Right-click Destinations and choose **Configure a new JMSQueue**.
3. In the right pane, enter `jms/REGISTRATION_MDB_QUEUE` in the Name field.

4. Enter `jms/REGISTRATION_MDB_QUEUE` in the JNDI Name field.
5. Click Create.
6. In the left pane, right-click Destinations and choose **Configure a new JMSQueue**.
7. In the right pane, enter `jms/MAIL_MDB_QUEUE` in the Name field.
8. Enter `jms/MAIL_MDB_QUEUE` in the JNDI Name field.
9. Click Create.
10. In the left pane, right-click Destinations and choose **Configure a new JMSQueue**.
11. In the right pane, enter `jms/XML_UPLOAD_MDB_QUEUE` in the Name field.
12. Enter `jms/XML_UPLOAD_MDB_QUEUE` in the JNDI Name field.
13. Click Create.
14. To verify that the queues were created:
    - Click the Destinations node in the left pane.
    - Verify that all three queues appear in the right-hand table.

Step 7: Add email capabilities to the MedRec application.

WebLogic Server includes the JavaMail API version 1.1.3 reference implementation from Sun Microsystems. Using the JavaMail API, you can add email capabilities to your WebLogic Server applications. To configure JavaMail for use in WebLogic Server, you create a Mail Session in the WebLogic Server Administration Console. A mail session allows server-side components and applications to access JavaMail services with JNDI, using Session properties you preconfigure for them.

1. In the left pane of the Administration Console, open Services.
2. Right-click the Mail node and choose **Configure a new MailSession**.
3. Enter:
    - `mail/MedRecMailSession` in the Name field.
    - `mail/MedRecMailSession` in the JNDIName field.

- `mail.user=joe;mail.host=mail.mycompany.com` in the Properties text field.

4. Click Create.
5. Select medreccluster in the Targets box.
6. Ensure that the All servers in the cluster option is selected.
7. Click Apply.
8. To verify that the factory was deployed:
   - Open the Clusters->medreccluster node in the left pane.
   - Select one of the servers in the cluster.
   - In the right pane, select the View JNDI tree link.
   - In the resulting window, open the mail node in the left pane.
   - Verify that `MedRecMailSession` appears in the left column under the mail node. This indicates that the session has been bound to the cluster JNDI tree.

Step 8: Configure the MedRec Sample Authenticator.

The MedRec Sample Authenticator retrieves login credentials from the configured PointBase RDBMS for a given username. Within the provider, passwords are validated, and if correct, the user's group associations are retrieved.

1. In the left pane of the Administration Console, expand the Security→Realms→myrealm→Providers node.
2. Select the Authentication node under the Providers node.
3. In the right pane, select Configure a New MedRec Sample Authenticator.
4. In the General tab in the right pane:
   - Enter MedRecSampleAuthenticator in the Name field.
   - Select SUFFICIENT from the Control Flag menu.
   - Click Create to create the new authenticator.

   The SUFFICIENT control flag indicates that the LoginModule does not need to succeed. If it does succeed, control is returned to the application. However, if it does not succeed, the server tries other configured authentication providers.

5. In the left pane of the Administration Console, expand the Security→Realms→myrealm→Providers→Authentication node.

6. Select the DefaultAuthenticator in the left pane of the Administration Console.

7. In the General tab in the right pane, select SUFFICIENT from the Control Flag menu.

8. Click Apply.

9. Because WebLogic Server cycles through available Authentication providers, reorder the provider list so that the PointBase database is not queried each time a login is attempted (for example, each time you log into the Administration Console in subsequent tutorials.

   Select the Security→Realms→myrealm→Providers→Authentication node in the left pane of the Administration Console.

10. In the right pane of the Console, select Re-Order the Configured Authentication Providers.

11. Use the arrows in the Configured Providers list to define the following order of authentication providers:

    a. DefaultAuthenticator

    b. MedRec Samples Authenticator

    c. DefaultIdentityAsserter

12. Click Apply.

Best Practices

- Only the JMS Server is pinned to a single server in the cluster. Target all other services to all servers in the cluster.
- When you create a JDBC DataSource, be sure you enable global transaction support (by checking the Honor Global Transactions box) so that your application can support transaction services.

The MedRec application uses a DataSource with transactions enabled (a `TxDataSource` object) when persisting application data to the database using entity beans.

- Consider using a persistent JMS store to ensure that your JMS messages are not lost if a problem occurs before the message-driven beans process the messages on the JMS queue.
- You cannot, however, configure a transaction (XA) JDBC connection pool to be used with a JMS JDBC store. JMS must use a JDBC connection pool that uses a non-XA resource driver. JMS does the XA support in the layer above the JDBC driver.

  This is because WebLogic JMS is its own resource manager. That is, JMS itself implements the XAResource and handles the transactions without depending on the database (even when the messages are stored in the database). This means that whenever you are using JMS and a database (even if it is the same database where the JMS messages are stored), then it is two-phase commit transaction.
- When configuring the persistent JMS store, you can persist JMS messages to a directory on the file system (called JMS file store) or to a database using JDBC (called JMS JDBC database store).   If you want better performance and do not need the reliability of a database, BEA recommends you persist JMS messages to the file system.

The Big Picture

The MedRec application uses JMS to create a new patient record. The asynchronous nature of JMS allows the task to be queued and completed later while the user continues with another task.

After the user clicks **Create** on the Web page to register a new patient, a JMS message is created and put on the `REGISTRATION_MDB_QUEUE` JMS queue. The `RegistrationEJB` message-driven bean takes the message off the queue and persists the new patient data to the database using an instance of the `PatientEJB` entity bean. The `PatientEJB` entity bean uses the JDBC DataSource to connect to the Pointbase database.

The MedRec application uses other entity beans to persist additional data to the database.

The MedRec application uses persistent JMS messaging, which means that the new patient JMS messages that are put on the queue are also stored in a Pointbase database so that the messages can be retrieved in case a problem occurs (such as a server crash) before the message-driven bean is able to process them. The application uses the JMS JDBC store to connect to and update the JMS tables in the Pointbase database.

Related Reading

- Introduction to WebLogic JDBC
- *Managing JDBC Connectivity*
- *WebLogic JMS Fundamentals*
- *JMS: Configuring*
- *Using JavaMail with WebLogic Server Applications*

1 Creating and Deploying to WebLogic Server Clusters

Tutorial 4: Deploying the MedRec Package to a Cluster

This tutorial describes how to deploy the MedRec applications to a cluster for production. The tutorial includes the following sections:

- Prerequisites
- Procedure
- Best Practices
- The Big Picture
- Related Reading

Prerequisites

Before starting this tutorial:

- Use the Configuration Wizard to create the clustered servers as described in Tutorial 1: Creating a WebLogic Server Cluster.
- Use the Administration Console to create and target MedRec application resources, as described in Tutorial 3: Setting Up Cluster Resources for MedRec.

Procedure

Now that your running cluster has the JDBC, JMS, and Mail resources required by the MedRec applications, you can deploy MedRec to the cluster.

- Step 1: Deploy the MedRec Application
- Step 2: Deploy the Physician Application
- Step 3: Test the Deployment

Step 1: Deploy the MedRec Application

1. Expand the Deployments node in the left pane of the console.
2. Right-click Applications and select Deploy a new Application.
3. In the right pane, use the links in the Location field to navigate to the `c:\bea\weblogic81\samples\server\medrec\dist` directory. After selecting this directory, the deployment assistant lists three MedRec applications as being available:
   - medrecEar
   - physicianEar
   - startupEar

   Note: The startupEar application is used to automatically start a browser and load the MedRec index page when you start a single-server MedRec domain. It is not intended for deployment to a cluster.
4. Select the medrecEar application.
5. Click Target Application.
6. In the Select targets for this application screen:

- Select medreccluster from the list of clusters.
- Ensure that the All servers in the cluster option is selected.
- Click Continue.

7. On the Review your choices and deploy page:
   - Select the "I will make the application accessible from the following location" option.
   - Click the Deploy button.
8. Monitor the deployment status for all components of medrecEar. When all modules have deployed, continue with the next step.

Step 2: Deploy the Physician Application

1. Expand the Deployments node in the left pane of the console.
2. Right-click Applications and select Deploy a new Application.
3. In the right pane, use the links in the Location field to navigate to the `c:\bea\weblogic81\samples\server\medrec\dist` directory. After selecting this directory, the deployment assistant lists three MedRec applications as being available:
   - medrecEar
   - physicianEar
   - startupEar

   Note: The startupEar application is used to automatically start a browser and load the MedRec index page when you start a single-server MedRec domain. It is not intended for deployment to a cluster.
4. Select the physicianEar application.
5. Click Target Application.
6. In the Select targets for this application screen:
   - Select medreccluster from the list of clusters.
   - Ensure that the All servers in the cluster option is selected.
   - Click Continue.

7. On the Review your choices and deploy page:
   - Select the “I will make the application accessible from the following location” option.
   - Click the Deploy button.
8. Monitor the deployment status for all components of medrecEar. When all modules have deployed, continue with the next step.

Step 3: Test the Deployment

To test whether the deployment was successful, open a browser window and enter the URL:

```
http://127.0.0.1:8001/start.jsp
```

You should receive the main login page where you can use the application from different roles.

Try accessing the same application using other server’s in the cluster by entering the URLs:

```
http://127.0.0.1:8011/start.jsp
```

```
http://127.0.0.1:8021/start.jsp
```

Each URL should provide the main login page.

Best Practices

Deploy applications to all members of the cluster (homogeneous deployment) to avoid JNDI conflicts. See Communications in a Cluster in Using WebLogic Server Clusters.

The Big Picture

In this tutorial, you specify the different port numbers in separate URL requests to access different members of the cluster. A more sophisticated development environment might use a single DNS name that maps to all addresses in the cluster. DNS cycles through the configured addresses in a round-robin fashion to distribute multiple requests to different IP addresses.

Production environments generally use load balancing hardware with sophisticated load algorithms and load monitoring to ensure that large volumes of requests are distributed network requests to servers evenly (or according to each machine's processing capacity).

Related Reading

- WebLogic Server Application Packaging
- Deploying Applications and Modules
- Load Balancing in a Cluster in *Using WebLogic Server Clusters*