Third Party Application
Using Liferay with Pramati Server
Pramati Server 3.5 SP6, 4.1, Liferay 2.2.1
Prepared by Development Team

1 Introduction

Liferay Portal provides cost-effective, personalizable tools that help you communicate and share information more efficiently. The primary advantages of using Liferay are:

- Provide organizations with a single sign-on web interface
- Allow administrators to easily manage users, groups, and roles through a GUI interface
- Give users personalization tools so they can modify their portal layout as they see fit.
- Deploy on J2EE servers or servlet containers running in multiple Java enabled operating systems.
- Enable pluggable databases

In this document, we will learn how to deploy Liferay on Pramati Server and access the portal.

2 Requirements

To use Liferay with Pramati Server:

- Download and Install Pramati Server 3.5 sp6 or higher from Pramati website. For this document, we will use Pramati Server 3.5 SP6.
- Download Liferay 2.2.1 (or higher) from http://sourceforge.net/project/showfiles.php?group_id=49260. For this document, we will use liferay-ep-2.2.1.ear.
- Use either Oracle or MySQL database. For this document, we will use Oracle database.
- Use JDK 1.4.x.

3 Preparing and Updating Liferay for Deployment

To prepare the application:

Create a directory called "modifier".

Move liferay-ep-2.2.1.ear to modifier directory.

Download and copy application.jar and Update_ear.bat to modifier directory.

Execute the bat file.

This procedure is necessary to update LoginAction.class in portal-ejb.jar, which performs the authentication in Pramati Server.

4 Configuring Pramati Server

Configure the following parameters in Server:

1. In runserver.bat of Server located in <$PSERVER-INSTALL>/server/bin/ folder, increase the JVM Size to -Xms256m -Xmx512m.
2. Apply the patch by copying the server_patch.jar (available on request) to <$PSERVER-INSTALL>\server\lib\pramati folder.

To start Pramati Server with node Liferay:

1. At the command prompt, go to <install_dir>\server\bin and type the following command:

   ```
   runserver -verbose -node liferay
   ```
2. Press Enter to create a node.
3. Enter the Naming Port as 9292.
4. Enter the HTTP Port as 8282.

Open deploy-config.xml located at <install_dir>\server\node\liferay\config\ and increase compilation memory as shown below:

Before:

```
<compilation>
         <out-of-process-compilation enabled="false">
           <!--external-compiler use="javac" path="C:\jdk1.3.101\bin\javac.exe" /-->
           <!--options>
                <property name="initial-heap-size-megabytes" value="120" />
                <property name="maximum-heap-size-megabytes" value="200" />
           </options-->
         </out-of-process-compilation>
</compilation>
```

After:

```
<compilation>
        <out-of-process-compilation enabled="true">
          <external-compiler use="javac" path="C:\j2sdk1.4.2_02\bin\javac.exe" />
          <options>
```

```
                <property name="initial-heap-size-megabytes" value="128" />
                <property name="maximum-heap-size-megabytes" value="512" />
            </options>
        </out-of-process-compilation>
    </compilation>
```

Note: For external-compiler, the attribute "path" should point to proper JDK installation.

Restart Server to reflect the above changes.

4.1 Creating a DataSource on Server

To create a JDBC DataSource using Management Console:

1. Click Configure > Resources.
2. In the Resources page, click Add under DataSources section.
3. Create New DataSource by entering the following details:
   - Driver: oracle
   - Class: Picked up automatically as oracle.jdbc.driver.OracleDriver
   - Connection URL: Picked up automatically as jdbc:oracle:thin:@<url>:1521:ORCL. Change it to jdbc:oracle:thin:@db:1521:db
   - Name: Liferay_CF
   - Classpath: Location of Driver classes. If not specified, Server picks it from the lib\ext folder.
   - Authorized by: Provide any username and password. For example, username Peter and password Mary.

Click Next. It is not necessary to provide any further details. Save the resource.

4.2 Creating Queues

To create queues:

1. Select Configure > Message Server > Queue.
2. In the New Queue Field, enter CacheQueue as the queue name. Retain all default parameters and click Add.
3. Use the same procedure as above to create a MailQueue.

4.3 Creating a Mail Session on Server

To create a mail session bound to mail/MailSession:

1. Select Configure > Message Server > Mail.
2. In the Mail Resources page, click Add.
3. Enter the following details:

- Name: MailSession
- Host: mail.yahoo.com
- Outgoing Email Address: mail.yahoo.com
- Username: Peter
- Password: Mary

4. Click Save.

4.4 Creating a Security Realm

To create a Security Realm:

1. Select Configure > Security.
2. Under the Realms section, click Add.
3. Type the realm Name as PortalRealm. Retain other default values and click Next.
4. Provide the complete path for the XML as C:\xml\temp\liferay_portal.xml
5. There is no need to add any other details. Save PortalRealm.

5 Deploying Liferay Application

To deploy the application:

1. Open Deploy Tool.
2. Use the Open button in Deploy Tool to browse and open liferay-ep-2.2.1.ear from the modifier directory.
3. Complete remaining tasks, if any.
4. Change the local JNDI name for the bean `com_liferay_portlet_documentlibrary_ejb_DLRepositoryLocalManager` to `ejb/liferay/DLRepositoryLocalManagerHome` as the lookup for this been is done using the reference name `ejb/liferay/DLRepositoryLocalManagerHome`. If it is not changed, all such references fail as it is a standard utility class and is not related to any bean or web component. On such failure, the JNDIUtil class in Liferay attempts to do a direct lookup by removing the `java:/comp/env`. For this to work, the reference name must be equal to the local JNDI name.

The EJB local ref mappings for web modules are automatically mapped to the remote JNDI names if the archive is opened in Deploy Tool. Change these manually to point to the local JNDI names:

1. Click on Web Properties > EJB.

2. Change the values of Link Name of all EJB-Ref Name. For example, change the reference for `ejb/liferay/AddressLocalManagerHome` to `ejb/liferay/AddressLocalManagerHome_PRAMATI_LOCAL`
3. Click the Deploy button.

To access the deployed application, access the following URL in your browser: http://<localhost>:8282/

6 Possible Errors and Workarounds

1. **java.lang.OutOfMemory**:

   Increase the JVM size in runserver.bat and deploy the application again.

2. **Class com.pramati.ejb.runtime.MailReceiptManagerHomeImpl_1367081944 not found**:

   Check if the values for compilation tag in deploy-config.xml have been set properly.