Installing Liferay DXP on Tomcat 8

Liferay DXP 7.0 bundled with Tomcat 8 is available on the Help Center (DXP) or the Community Downloads page (Portal CE). The Tomcat bundle contains JARs, scripts, and configuration files required for deploying Liferay DXP 7.0. Copying these files from the Liferay DXP Tomcat bundle facilitates installing Liferay DXP on an existing Tomcat application server.

Whether you copy bundle files (recommended) or download and create the files, you must download these *Additional Files* for DXP or Portal CE:

- Liferay DXP WAR file
- Dependencies ZIP file
- OSGi Dependencies ZIP file

Installing Liferay DXP manually requires these basic steps:

- Installing Liferay DXP dependencies to your application server
- Configuring your application server for Liferay DXP
- Installing the Liferay DXP WAR file to your application server

You'll see the term *Liferay Home* used in this installation guide. *Liferay Home* refers to the folder containing your Tomcat server folder. When Liferay DXP is installed on Tomcat, the Liferay Home folder contains the Tomcat server folder as well as `data`, `deploy`, `license`, and `osgi` folders. You'll also see the terms `$CATALINA_HOME` and `$CATALINA_BASE` used in this guide. `$CATALINA_HOME` refers to the folder that contains Tomcat's binaries, while `$CATALINA_BASE` refers to the folder that contains your configuration–just as you're used to from Tomcat's generic installation. If you download a Liferay Tomcat-bundle, both refer to the same folder (just as with a standard Tomcat installation). This folder is usually named `tomcat-[version]` or `apache-tomcat-[version]`.

Installing Liferay DXP Dependencies

Liferay DXP depends on many JARs that are included in the Liferay DXP Tomcat bundle. Some JARs in the bundle are not strictly required but can still be useful. If you don't have a Liferay DXP Tomcat bundle, you can download

1. If you downloaded a Liferay DXP Tomcat bundle, extract the bundle to a temporary location of your choosing. You'll copy a number of resources from this bundle to your Tomcat server as you manually install Liferay DXP.

2. If you have a Liferay DXP Tomcat bundle, copy all the JARs from your bundle's `$CATALINA_BASE/lib/ext` folder to your application server's `$CATALINA_BASE/lib/ext` folder. If the `$CATALINA_BASE/lib/ext` folder doesn't exist on your application server, create it. If you don't have a Liferay DXP Tomcat bundle, you'll have to individually download the JARs listed below.

   Here's a list of the JARs that you need to copy or download to your `$CATALINA_BASE/lib/ext` folder:

   - `activation.jar` - http://www.oracle.com/technetwork/java/jaf11-139815.html
   - `ccpp.jar` - http://mvnrepository.com/artifact/javax.ccpp/ccpp/1.0
   - `com.liferay.registry.api.jar` - https://repository.liferay.com/nexus/content/repositories/liferay-public-releases/com/liferay/com.liferay.registry.api
   - `hsql.jar` - http://hsqldb.org/doc/src/org/hsqldb/jdbc/JDBCDriver.html
   - `jms.jar` - http://www.oracle.com/technetwork/java/docs-136352.html
   - `jta.jar` - http://www.oracle.com/technetwork/java/javaee/jta/index.html
   - `jutf7.jar` - http://sourceforge.net/projects/jutf7
   - `mail.jar` - http://www.oracle.com/technetwork/java/index-138643.html
   - `mysql.jar` - http://dev.mysql.com/downloads/connector/j
   - `persistence.jar` - http://www.oracle.com/technetwork/java/javaee/tech/persistence-jsp-140049.html
   - `portal-kernel.jar` - http://mvnrepository.com/artifact/com.liferay.portal/com.liferay.portal.kernel
   - `portlet.jar` - http://mvnrepository.com/artifact/javax.portlet/portlet-api
   - `postgresql.jar` - https://jdbc.postgresql.org/download.html
   - `support-tomcat.jar` - http://repo1.maven.org/maven2/com/liferay/portal/support-tomcat
3. Make sure that Tomcat can access the JDBC driver for your database. The list of JARs above includes `mysql.jar` and `postgresql.jar`. If you're using a database whose JDBC driver is not included in the list above, download the driver and copy it to your `$CATALINA_BASE/lib/ext` folder.

4. Create an `osgi` folder in your Liferay Home folder. Then extract the OSGi ZIP file that you downloaded into the `osgi` folder.

   Liferay DXP requires an OSGi runtime, and the `osgi` folder provides this with many required JAR files and configuration files.

1. You should have the following files in the `$CATALINA_BASE/lib/ext` folder:

    - `activation.jar`
    - `ccpp.jar`
    - `com.liferay.registry.api.jar`
    - `hsql.jar`
    - `jms.jar`
    - `jta.jar`
    - `jutf7.jar`
    - `mail.jar`
    - `mysql.jar`
    - `persistence.jar`
    - `portal-kernel.jar`
    - `portlet.jar`
    - `postgresql.jar`
    - `support-tomcat.jar`
2. The `osgi` folder has the following subfolders:

    - `configs`
    - `core`
    - `marketplace`
    - `target-platform`
    - `test`

Tomcat Configuration

Next, you need to configure Tomcat for running Liferay DXP.

1. If you're working with a bundle, copy the `setenv.bat` and `setenv.sh` files from your bundle to your `$CATALINA_BASE/bin` folder. If not, create these files.

    These files set a number of JVM options for Catalina, which is Tomcat's servlet container. Among these options is the location of the Java runtime environment. If this environment is not available on your server globally, you must set its location in this file so Tomcat can run. Do this by pointing the `JAVA_HOME` environment variable for your OS to the location of the Liferay DXP supported JRE:

    ```
    export JAVA_HOME=/usr/lib/jvm/java-8-jdk
    ```

Once you've done this, configure Catalina's options to support Liferay DXP:

```
CATALINA_OPTS="$CATALINA_OPTS -Dfile.encoding=UTF-8 -Djava.net.preferIPv4Stack=tr
```

This sets the file encoding to UTF-8, prefers an IPv4 stack over IPv6, prevents Tomcat from working around garbage collection bugs relating to static or final fields (these bugs don't exist in Liferay DXP and working around them causes problems with the logging system), sets the time zone to GMT, and gives the JVM 1GB of RAM.

Important: For Liferay DXP to work properly, the application server JVM must use the `GMT` time zone and `UTF-8` file encoding.

These are initial settings. After installation you should tune your system for performance. As a result of that process, you may find you want to change particularly the amount of RAM available to Liferay DXP based on how your system runs.

2. If you're working with a bundle, copy the `$CATALINA_BASE/conf/Catalina/localhost/ROOT.xml` file from your bundle to the corresponding location in your application server. If not, create this file. The `ROOT.xml` file creates a web application context for Liferay DXP. `ROOT.xml` looks like this:

```
<Context path="" crossContext="true">

    <!-- JAAS -->

    <!--<Realm
        className="org.apache.catalina.realm.JAASRealm"
        appName="PortalRealm"
        userClassNames="com.liferay.portal.kernel.security.jaas.PortalPrincipal"
        roleClassNames="com.liferay.portal.kernel.security.jaas.PortalRole"
    />-->

    <!--
    Uncomment the following to disable persistent sessions across reboots.
    -->

    <!--<Manager pathname="" />-->

    <!--
    Uncomment the following to not use sessions. See the property
```

```
    <!--<Manager className="com.liferay.support.tomcat.session.SessionLessManagerI
</Context>
```

Setting `crossContext="true"` allows multiple web applications to use the same class loader. The configuration above includes commented instructions and tags for configuring a JAAS realm, disabling persistent sessions, and disabling sessions entirely.

3. Next, you should make sure that the libraries you copied to `$CATALINA_BASE/lib/ext` are loaded when you start your server. If you're working with a bundle, copy the `$CATALINA_BASE/conf/catalina.properties` file from your bundle to your server. If not, open `$CATALINA_BASE/conf/catalina.properties` and replace the line

```
common.loader=${catalina.base}/lib,${catalina.base}/lib/*.jar,${catalina.home}/li
```

with this one:

```
common.loader="${catalina.base}/lib","${catalina.base}/lib/*.jar","${catalina.hom
```

This allows Catalina to access the JARs that you copied to `$CATALINA_BASE/lib/ext` .

4. If you're working with a bundle, copy the `$CATALINA_BASE/conf/catalina.policy` file from your bundle to your server. If not, just replace the contents of the `$CATALINA_BASE/conf/catalina.policy` file with this:

```
grant {
    permission java.security.AllPermission;
};
```

If you want to enable PACL for Liferay DXP, you have to enable Tomcat's security manager and instruct Catalina to use the `$CATALINA_BASE/conf/catalina.policy` file. See the Enabling PACL section for more information.

5. Next, you should make sure that UTF-8 URI encoding is used consistently. If you're working with a bundle, copy the `$CATALINA_BASE/conf/server.xml` file to your server. If not, you can simply make a few edits to `server.xml` . Edit your `$CATALINA_BASE/conf/server.xml` file and add the attribute `URIEncoding="UTF-8"` wherever you see `redirectPort=8443` , in the definition of your connectors (HTTP and AJP). For example:

should become

```
<Connector port="8080" protocol="HTTP/1.1" connectionTimeout="20000" redirectPort=
```

And

```
<Connector port="8009" protocol="AJP/1.3" redirectPort="8443" />
```

should become

```
<Connector port="8009" protocol="AJP/1.3" redirectPort="8443" URIEncoding="UTF-8"
```

6. If you're on Unix, Linux, or Mac OS, navigate to your `$CATALINA_HOME/bin` and `$CATALINA_BASE/bin` folders and run the following command

```
chmod a+x *.sh
```

This command makes the shell scripts in Tomcat's `bin` folder executable.

Checkpoint:

At this point, you've finished configuring the application server's JVM settings.

1. The file encoding, user time-zone, preferred protocol stack have been set in the `setenv.sh` (`setenv.bat` for Windows).
2. The default amount of memory available has been increased.
3. The web application context has been declared in `$CATALINA_BASE/conf/Catalina/localhost/ROOT.xml` .
4. The `common.loader` property which allows Catalina to access the JARs in `$CATALINA_BASE/lib/ext` has been updated in `$CATALINA_BASE/conf/catalina.properties` .
5. All Java permissions have been granted in the `$CATALINA_BASE/conf/catalina.policy` file.

7. If in a Unix/Linux environment, the `chmod a+x *.sh` command has been run to the shell scripts in Tomcat's `bin` folder executable.

Tomcat Database Configuration

The easiest way to handle your database configuration is to let Liferay DXP manage your data source. If you want to use Liferay DXP's built-in data source, you can skip this section. When you first Liferay DXP, you can enter the required database configuration information on the Basic Configuration page.

If you want Tomcat to manage your data source, use this procedure:

1. Make sure your database server is installed and working. If it's installed on a different machine, make sure it's accessible from your Liferay DXP machine.

2. Add your data source as a resource in the context of your web application specified in `$CATALINA_BASE/conf/Catalina/localhost/ROOT.xml`:

```
<Context...>
    ...
    <Resource
        name="jdbc/LiferayPool"
        auth="Container"
        type="javax.sql.DataSource"
        driverClassName="com.mysql.jdbc.Driver"
        url="jdbc:mysql://localhost/lportal?useUnicode=true&amp;characterEncoding=
        username="root"
        password="root"
        maxTotal="100"
        maxIdle="30"
        maxWaitMillis="10000"
    />
</Context>
```

Note that the above resource definition assumes your database name is *lportal*, that you're using MySQL, and that your MySQL username and password are both *root*. You'll have to update these values with your own database name and credentials.

Your Tomcat managed data source is now configured. Next, let's consider your mail session.

Tomcat Mail Configuration

As with database configuration, the easiest way to handle mail configuration is to let Liferay DXP handle your mail session. If you want to use Liferay DXP's built-in mail session, skip this section and use Liferay DXP's Control Panel to configure a mail server after Liferay DXP has been installed and started.

If you want to manage your mail session with Tomcat, use these instructions:

To create a mail session bound to `mail/MailSession`, edit `$CATALINA_BASE/conf/Catalina/localhost/ROOT.xml` and configure your mail session. Make sure to replace the example mail session values with your own.

```
<Context...>
    <Resource
        name="mail/MailSession"
        auth="Container"
        type="javax.mail.Session"
        mail.pop3.host="pop.gmail.com"
        mail.pop3.port="110"
        mail.smtp.host="smtp.gmail.com"
        mail.smtp.port="465"
        mail.smtp.user="user"
        mail.smtp.password="password"
        mail.smtp.auth="true"
        mail.smtp.starttls.enable="true"
        mail.smtp.socketFactory.class="javax.net.ssl.SSLSocketFactory"
        mail.imap.host="imap.gmail.com"
        mail.imap.port="993"
        mail.transport.protocol="smtp"
        mail.store.protocol="imap"
    />
</Context>
```

Your mail session is configured. Next, you'll make sure Liferay DXP can access your mail session and database.

Configuring Tomcat-managed Database and Mail Sessions

1. If you will use *Liferay DXP* to manage your data source, simply follow the instructions on the Basic Configuration page that appears when you first start Liferay DXP.

   If you are using *Tomcat* to manage your data source, add the following line to your `portal-ext.properties` file in your *Liferay Home* folder to specify your data source:

   ```
   jdbc.default.jndi.name=jdbc/LiferayPool
   ```

2. If you will use *Liferay DXP* to manage your mail session, you can configure the mail session once Liferay DXP has started. That is, after starting your portal as described in the *Deploying Liferay DXP* section, go to *Control Panel* → *Server Administration* → *Mail* and enter the information required to configure your mail session.

   If you are using *Tomcat* to manage your mail session, add the following configuration to your `portal-ext.properties` file to reference that mail session:

   ```
   mail.session.jndi.name=mail/MailSession
   ```

It's just that easy! Before you deploy Liferay DXP, you should configure Portal Access Control Language (PACL) with Liferay DXP on Tomcat.

Enabling PACL

To enable PACL, you need to enable Tomcat's security manager. In the Tomcat Configuration section above, you already added the required permissions to the Tomcat policy configuration file, `catalina.policy`.

- Edit your `$CATALINA_BASE/bin/setenv.sh` (if on Linux, Unix, or Mac OS) or `setenv.bat` (if on Windows) and enable the security manager by inserting the following code into the `CATALINA_OPTS` variable (inside the quotation marks):

  ```
  -Djava.security.manager -Djava.security.policy=$CATALINA_BASE/conf/catalina.policy
  ```

- Check that your `$CATALINA_BASE/conf/catalina.policy` file specifies the required permissions (you should have already addressed this in the Configuring Tomcat section):

  ```
  grant {
      permission java.security.AllPermission;
  ```

To enable the security manager on Tomcat, the server must be started with the `-security` command line option. Shut down your Tomcat instance and then restart it with the following command:

```
./startup.sh -security
```

Tomcat reports the message `Using Security Manager` to your terminal.

Now you have PACL enabled and configured for your portal.

Deploying Liferay DXP

Now you're ready to deploy Liferay DXP using your Liferay DXP WAR file.

1. If you are manually installing Liferay DXP on a clean Tomcat server, delete the contents of the `$CATALINA_BASE/webapps/ROOT` directory. This removes the default Tomcat home page.

2. Extract the Liferay DXP `.war` file to `$CATALINA_BASE/webapps/ROOT`, so that a `WEB-INF` folder, among other files, is in there.

   Now it's time to launch Liferay DXP on Tomcat!

3. Start Tomcat by navigating to `$CATALINA_HOME/bin` and executing `./startup.sh` or `startup.bat`. Alternatively, you can use `./catalina.sh run` or `catalina.bat run`. Using one of the latter commands makes your terminal or command prompt tail Liferay DXP's log file. This can be useful if you want to see the startup activities Liferay DXP performs or debug deployment.

Congratulations on successfully installing and deploying Liferay DXP on Tomcat!

« Installing Liferay DXP Manually Installing Liferay DXP on Wildfly »

¿Fue útil este artículo?

Sí No

Usuarios a los que les pareció útil: 0 de 1

¿Necesitas más ayuda?

Abre un ticket

Contacta con nosotros

Pregunta en la Comunidad

Liferay University
Forma a tu equipo en los puntos esenciales del uso de Liferay con videos detallados y ejercicios prácticos diseñados para simular una formación presencial

Liferay Global Services
Aumenta tu conocimiento sobre el producto y el desarrollo gracias a expertos que te ayudarán a maximizar tu ROI