Upgrade Liferay 6.2 installation to Liferay 7 in simple steps

As we have studied so far, Liferay 7 is much exciting release from Liferay Inc. and certainly it is a promising product that will have many new functional features and many architectural enhancements or rather improvements. This section will cover the gist about How to upgrade Liferay 6.2 to Liferay 7:

Fixing database (If required)
Setting up the DXP
Managing Extra Settings
Running the Upgrade utility
Fixing Memory Issue

In this section, we are going to upgrade a Liferay 6.2 installation to Liferay 7. Before you perform an upgrade, make sure you have complete back up of your existing installation. This backup might include –

- Database backup
- Data folder backup

The steps mentioned in this section are performed on a Liferay instance which was used as an HRMS platform. This has portlets deployed for employee management, project and allocation management, timesheet, trainings management, performance management, leave management, exit process module etc. All of the portlets use the Liferay database for their services and are heavily dependent on Liferay's entities. Database used is MySQL. This installation uses the default data directory used by Liferay.

Fixing your database, just in-case

Before we start the process, make a copy of your database. We will be using this database for our upgrade process. Make sure your database charset is UTF-8. If the database is not properly set to UTF-8, you might face the exception similar to the one below –

```
com.liferay.portal.kernel.upgrade.UpgradeException:
com.liferay.portal.kernel.upgrade.UpgradeException: java.sql.SQLException: Incorrect
string value: '\xEC\x84\xB8\xEA\xB3\x84...' for column 'name' at row 4

        at
```

```
com.liferay.portal.kernel.upgrade.UpgradeProcess.upgrade(UpgradeProcess.java:91)

        at
com.liferay.portal.kernel.upgrade.util.UpgradeProcessUtil._upgradeProcess(UpgradeProcessUtil.java:175)

        at
com.liferay.portal.kernel.upgrade.util.UpgradeProcessUtil.upgradeProcess(UpgradeProcessUtil.java:143)

        at
com.liferay.portal.kernel.upgrade.util.UpgradeProcessUtil.upgradeProcess(UpgradeProcessUtil.java:125)

        at
com.liferay.portal.events.StartupHelper.upgradeProcess(StartupHelper.java:164)

        at
com.liferay.portal.events.StartupHelperUtil.upgradeProcess(StartupHelperUtil.java:81)

        at com.liferay.portal.tools.DBUpgrader.upgrade(DBUpgrader.java:157)

        at com.liferay.portal.tools.DBUpgrader.main(DBUpgrader.java:103)
```

To fix your database, you can run alter queries on your tables. Your query on a table would look like –

```
ALTER TABLE lportal.Account_ CHARACTER SET utf8 COLLATE utf8_general_ci;
ALTER TABLE lportal.Account_ CONVERT TO CHARACTER SET utf8 COLLATE utf8_general_ci;
```

In the queries above, we are changing Account_ table and the database name is lportal. You need to run similar queries for all tables in your database.

Writing queries for each table can be time consuming and not encouraged. You can utilize a script to generate the queries. This script will generate all of the queries and put in a text file of your choice. After running the script, you can run the queries from the file to fix your database. This script is shared on Liferay forum post here - https://web.liferay.com/community/forums/-/message_boards/message/76256549

The script is –

```
SELECT CONCAT ("ALTER TABLE ", TABLE_SCHEMA,".",TABLE_NAME," CHARACTER SET utf8
COLLATE utf8_general_ci;   ",
"ALTER TABLE ", TABLE_SCHEMA,".",TABLE_NAME," CONVERT TO CHARACTER SET utf8
COLLATE utf8_general_ci;  ") AS alter_sql FROM information_schema.TABLES WHERE
TABLE_SCHEMA = 'lportal _1' INTO OUTFILE '/Users/ravi.gupta/sql/fix-utf-lportal.txt';
```

This script will store the queries to fix-utf-lportal.txt file to respective location. Running these queries will fix your database.

Now your database is ready for upgrade.

Setting up the DXP

To perform an upgrade, we will be using a fresh liferay-7 tomcat bundle. The version we are going to use is 7.0 community edition, GA3. In the extracted bundle, there is a directory named

tools. This has another directory *portal-tools-db-upgrade-client* inside which has all necessary stuff we need for upgrade process.

/tools

/portal-tools-db-upgrade-client

- app-server.properties
- com.liferay.portal.tools.db.upgrade.client.jar
- portal-upgrade-database.properties
- portal-upgrade-ext.properties

In the upgrade client directory, we have properties files for db upgrade, app server and for upgrade ext along with a jar file which is actually a client jar we need to run to begin upgrade process. This is unlike previous versions of Liferay, where upgrade process will automatically begin when we start the new instance of Lifeary. Starting from v7.0, we need to manually begin the upgrade process.

Let's update the properties file so that upgrade utility can pick properties correctly and then utilize them to upgrade Liferay.

app-server.properties

This file takes 4 properties which are used by utility to know the path of the Liferay server instance and library path. These properties are –

Property	Description	Optional
dir	Path of the application server used by Liferay server. In case of tomcat it usually is - /path/to/liferay/liferay-ce-portal-7.0-ga3/tomcat-8.0.32	No
portal.dir	Path to the portal deployment directory. In case of tomcat and if you have Liferay deployed on ROOT then – tomcat-8.0.32/webapps/ROOT	No
global.lib.dir	Global lib directory as per the application server used. In case of tomcat, it is - tomcat-8.0.32/lib	No
extra.lib.dir	Comma separated list of all library paths which we want to include to classpath. In case of tomcat one of the dir can be - tomcat-8.0.32/lib/ext	No

If your Liferay instance is located at - /Users/ravi.gupta/liferay-ce-portal-7.0-ga3 then your app-server.properties file would look like –

```
dir=/Users/ravi.gupta/liferay-ce-portal-7.0-ga3/tomcat-8.0.32
global.lib.dir=lib
portal.dir=webapps/ROOT
extra.lib.dirs=/Users/ravi.gupta/liferay-ce-portal-7.0-ga3/tomcat-8.0.32/lib/ext
```

Yup, this takes relative paths to your tomcat directory.

portal-upgrade-database.properties

This file helps upgrade utility to know the database connection related values. The table below lists the properties which go into this file –

Property	Description	Optional
jdbc.default.url	Jdbc connection string	No
jdbc.default.driverClassName	Driver class name. In case of mysql - com.mysql.jdbc.Driver	No
jdbc.default.username	Database connection user name	No
jdbc.default.password	Database connection passwrod	No

The properties file would look like –

```
jdbc.default.url=jdbc:mysql://localhost/lporta?characterEncoding=UTF-8&dontTrackOpenResources=true&holdResultsOpenOverStatementClose=true&useFastDateParsing=false&useUnicode=true
jdbc.default.driverClassName=com.mysql.jdbc.Driver
jdbc.default.username=ravi
jdbc.default.password=secret
```

These are the same properties file as used in portal-ext.properties file.

portal-upgrade-ext.properties

This file contains portal properties which are to be utilized during upgrade process. Only one property is required to be added as listed in the table below –

Property	Description	Optional
liferay.home	The path which points to liferay home directory.	No
hibernate.jdbc.batch_size	JDBC bath size. This defaults to 250.	Yes

This file would look like below. Only one property needed.

liferay.home=/Users/ravi.gupta/liferay-ce-portal-7.0-ga3

It's a good practice to update these properties files before we start upgrade. If, however, we do not update these files, you will still be able to begin the upgrade process and upgrade utility will prompt you to provide correct information at the start.

Extra settings

Usually the three properties files we just discussed are necessary for upgrade but there are few other things as well which you must consider before starting the upgrade process.

Indexing

First thing is whether you should disable the indexing during upgrade. To avoid any indexing issues and performance issues, disabling indexing is a good option. This will ensure that indexer does not run during upgrade and no issue will occur related to indices.

To disable indexing, create a new file named

com.liferay.portal.search.configuration.IndexStatusManagerConfiguration.cfg in liferay-home/osgi/configs directory. We need just one property, indexReadOnly, in this file with value as true. The file content would look like –

indexReadOnly=true

This will disable the indexing but after the upgrade you should enable indexing and then perform a full re-indexing from control panel.

Modules upgrade

Next is whether we want to upgrade the modules automatically or manually. To disable auto upgrade, create a file named com.liferay.portal.upgrade.internal.configuration.ReleaseManagerConfiguration.cfg in liferay-home/osgi/configs directory. The content of the file would be –

autoUpgrade=false

This will disable the auto upgrading of the modules.

Running the Upgrade utility

Now, we are all set to begin the database upgrade. Start the utility by running –
java –jar com.liferay.portal.tools.db.upgrade.client.jar

This will begin the upgrade process. In the logs, you would notice usual logs similar to when we start Liferay 7 instance. After few seconds, you will find upgrade logs starting with –

```
INFO [main][UpgradeProcess:82] Upgrading com.liferay.portal.upgrade.UpgradeProcess_7_0_0
INFO [main][UpgradeProcess:82] Upgrading com.liferay.portal.upgrade.v7_0_0.UpgradeSchema
...
```

During the process, it will update the database tables and indices. Finally, you will see a log entry for successful core upgrade –

Completed Liferay core upgrade and verify processes in 482 seconds

This notifies that the Liferay core has been successfully upgraded and if yes, then you will be promted with gogo shell –

You are connected to Gogo shell.

Upgrade commands:
exit or quit - Exit Gogo Shell
upgrade:check - List upgrades that have failed, have not started, or are still running
upgrade:execute {module_name} - Execute upgrade for specified module
upgrade:help - Show upgrade commands
upgrade:list - List registered upgrades
upgrade:list {module_name} - List upgrade steps required for specified module
upgrade:list | grep Registered - List registered upgrades and their current version
upgrade:list | grep Registered | grep steps - List upgrades in progress
verify:execute {module_name} - Execute verifier for specified module
verify:list - List registered verifiers

You will be able to run commands on the shell now. Try upgrade:check command to check if any component failed to upgrade.

Fixing memory issue

In case, you face any memory related error, you should consider changing memory settings. By default, Java memory is set to 2GB and you can update it to 4GB or more. To update the properties, you can use –j flag and then supply the settings –

java -jar com.liferay.portal.tools.db.upgrade.client.jar -j "-Dfile.encoding=UTF8 -Duser.country=US -Duser.language=en -Duser.timezone=GMT -Xmx4096m"

This will help you to prevent any memory related issue.

UNITED STATES

Azilen Technologies LLC
6476, Orchard Lake Road,
Ste D, West Bloomfield,
MI 48325

Tel: +1-972-325-2243

BELGIUM

Azilen Technologies BVBA
Diestseweg 32C,
2440 Geel,
Belgium

Tel: +32 3 8082588

INDIA

Azilen Technologies Pvt. Ltd.
404/405,Iscon Mall,
Near Shivranjani Cross Road,
Satellite, Ahmedabad-380015

Tel: +91-79 4009 3121

Email : info@azilen.com | Website : www.azilen.com