Upgrading From Liferay DXP 7.x to 7.4

Table of Contents

The DXP Upgrade Process

If you have upgraded from prior versions of Liferay DXP, you will know that upgrades can be a time-consuming and resource-intensive process. However, from Liferay DXP 7.1 and onwards, we have introduced several features that make things easier. Customers who have upgraded from 6.2 to 7.x can be assured that these changes will make the task of upgrading easier than it was in the past.

This paper is a high-level overview of the effort involved in moving from Liferay DXP 7.x to 7.4. For an in-depth look at the process, please review the Liferay documentation.

Breaking Changes to Liferay DXP 7.4

We fully document the breaking changes in new versions of Liferay from the version before. These changes might affect the functionality of certain modules. For more information about API changes that could affect developed modules, please review the Breaking Changes list. This list is updated regularly.

Upgrade Tooling

One of the new features introduced in Liferay DXP 7.1, and available in Liferay DXP 7.4, is the ability to restart an upgrade. In prior versions, if an upgrade failed, the database had to be restored and the whole process would begin again. With the new upgrade tool, there is no need to restore from a backup. Simply address the error that stopped the upgrade and resume it.

Upgrade on Startup

For Liferay DXP 7.4, the auto-upgrade behavior can be controlled by a new portal property which covers both the Core and Module upgrades:

- upgrade.database.auto.run=true

The default value for this property is `false` so upgrades will not run automatically on startup to provide users more control over database changes.

This property can be especially useful for container environments where an upgrade to the database is needed on startup. For this use case, just set the property to `true`. Be sure to treat this startup as you would any regular upgrade, performing needed measures like making a backup, as well as any other pre-upgrade tasks.

Liferay now releases Updates instead of Service Packs and Fix Packs. However, this property is still relevant when upgrading bundles and also in regards to applying hotfixes supplied by Support. Updates contain fixes to known issues as well as new features. Customers with this property in place should carefully review the details of an Update before applying it, and apply the Update first in a non-Production environment. Please read the following article to get more information about it.

Upgrade Checklist

The following is a brief overview of the necessary steps required prior to starting an upgrade. These are the main points of preparation and general tasks that should be done to ensure a successful upgrade.

JDK

Liferay DXP 7.4 supports the use of Java JDK 11 for runtime and project compilation. Liferay recommends the use of JDK 11, however for customers with extended support contracts, JDK 8 is also supported. There are numerous flavors of JDKs out there, and as long as they are Java Technical Compatibility Kit (TCK) compliant builds, they are fully supported with Liferay DXP 7.4.

Back Up and Make Copies of Everything

Before getting started, it is highly recommended that you back up of everything and restore it to a separate environment. This ensures the testing of restore procedures to ensure they are up to date and correct — and you don't need to change your production system before you're sure that the upgrade process runs smoothly. For a Liferay project, that is usually the database, document repository and application server. For some application servers, a backup of the entire app server can be made (Tomcat, Wildfly, etc.) whereas other app servers cannot be copied in full. Refer to your specific app server documentation for details on how backups can be taken.

The backup process preserves any deployed applications and plugins, all data and all documents in the Liferay system in a manner that would allow for a quick restoration of the data if needed for any reason.

Modules

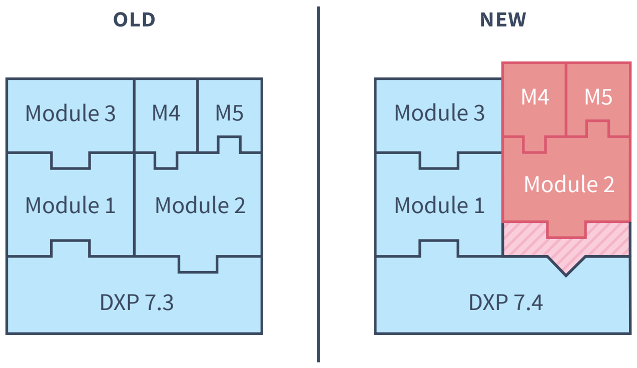
The good news about upgrading from Liferay DXP 7.x to 7.4 is that the big hump is behind you. There were substantial architectural changes from 6.2 to 7.0, as Liferay Portal moved from the monolithic approach to the much more modular, OSGi approach of Liferay DXP. This required a rewriting of almost all portlets and plugins to convert to modules.

If you have already done the conversion to modules, please cross-reference any custom modules with the Breaking Changes list to ensure they continue to work properly.

If you have existing modules in 7.0, 7.1, 7.2, or 7.3 deployed, they can be separated into two kinds of modules: Liferay Marketplace Apps and custom build modules. If you are using any Marketplace apps, check the Liferay Marketplace for an updated version of the app.

Modules that you have built will need to be at the very least recompiled against a Liferay DXP 7.4 workspace for them to function. Changes in Liferay APIs may cause modules to cease to function properly, so don't forget to consult the Breaking Changes list linked above.

Liferay now provides a tool to check if module versions have changed between versions of Liferay DXP. This is useful if a custom module is making use of a Liferay module and to ensure that things still work across versions. You can access the tool here.



Above: By means of example, in a Liferay DXP 7.3 application, all the modules fit together and work fine, but in 7.4, one of the extension points (APIs) has changed, so if taking Module 2 as-is from 7.3 and deploying it into 7.4 causes it to break, then any subsequent modules that depend on it, such as Module 4 and Module 5, will also break.

Headless APIs

New in Liferay DXP 7.2 was a commitment to a fully headless system, using APIs written using the OpenAPI standard. Moving forward, those APIs should not break between versions.

Themes

In Liferay DXP 7.0, themes were built with Bootstrap 3. The 7.1 release moved to Bootstrap 4. In an effort to keep backwards compatibility, we have introduced a Compatibility Layer for Bootstrap. Themes written with Bootstrap 3 will attempt to be compatible with Bootstrap 4, so you won't have to rewrite the whole theme. However, the Compatibility Layer may not be perfect, so you may experience some issues.

Liferay DXP 7.4 continues using Bootstrap 4 (version 4.4.1), the same as 7.3, 7.2, and 7.1. If you are coming from 7.1, 7.2, or 7.3, you will not need to migrate Bootstrap versions. However, you may need to update your module requirements and metadata. As with custom modules, these should be recompiled against a 7.2 workspace.

With the change to Bootstrap 4, you may prefer to rewrite your theme. The Compatibility Layer exists so that you don't have to do this immediately and gives you options to apply fixes so your theme is compatible with Bootstrap 4 until you are ready to rewrite. For further information, refer to the Stylebooks section below.

For more information about the Bootstrap compatibility layer, please review the documentation.

Document Repository

While there are no major changes to the document repository from Liferay DXP 7.x to 7.4, you should always make sure you backup the document repository file system and double check that the path to the repository is correct. By default, Liferay DXP stores the file system in `[Liferay Home]/data/document_library` of each DXP bundle or installation. When upgrading, you can copy the old document repository to the new location, leaving the old bundle to function as a rollback point.

It is possible that the liferay.home property is defined somewhere, such as `portal-setup-wizard.properties` or `portal-ext.properties` file. When migrating to a new server, double-check that the path defined in the property is correct in the new environment.

If you are using a Store type other than FileSystemStore, don't forget to make the change in `portal-ext.properties` in the new Liferay DXP deployment.

Search

With Liferay DXP 7.0, we introduced Elasticsearch as the default search engine, moving from Lucene, and introduced additional search capabilities based on Elasticsearch with Liferay Enterprise Search. Elasticsearch is known for their rapid release cadence, and frequently retire old versions and release new versions. If you are using Elasticsearch, and you're not on a supported version, please take the time to update your Elasticsearch deployments to supported versions.

If you're upgrading to Liferay DXP 7.4 and are currently using Solr, you will need to move to Elasticsearch. Solr compatibility has been removed and is not supported with 7.4. Elasticsearch is required.

Each version of Liferay DXP supports different versions of Elasticsearch, which as noted above do change rapidly. Please refer to the Liferay Search Engine Compatibility Matrix for full details.

If you are using Liferay Enterprise Search, prior versions required the deployment of an Elasticsearch connector. That is no longer required and full functionality is bundled as a part of the platform.

Infrastructure

Given that there is a new version of Liferay DXP, your company may be considering changing your infrastructure. New servers? Going to the cloud? Going on-premise?

Liferay DXP 7.4 has an updated Compatibility Matrix that outlines the supported versions of various Application Servers, Databases and Browsers. If you're coming from 7.x and going to 7.4, the supported versions won't have significant changes and there may be some overlap.

Sample setup showing supported versions in 7.0 vs 7.1 vs 7.2 vs 7.3 vs 7.4. Note that older versions in 7.1 are not supported.

Liferay DXP	**DXP 7.0**	**DXP 7.1**	**DXP 7.2**	**DXP 7.3**	**DXP 7.4**
OS	CentOS 6, CentOS 7	CentOS 7	CentOS 7	CentOS 7, CentOS 8	CentOS 7, CentOS 8
Application Server	Tomcat 8.0, Tomcat 8.5	Tomcat 9.0	Tomcat 9.0	Tomcat 9.0	Tomcat 9.0
Database	MySQL 5.6, MySQL 5.7	MySQL 5.7	MySQL 5.7, MySQL 8	MySQL 5.7, MySQL 8.0	MySQL 5.7, MySQL 8.0

You can find more information on supported versions and technologies in the updated Liferay DXP 7.4 Compatibility Matrix.

Database

For the database, the main task is to back it up and create a duplicate database by restoring the backup. This tests the backup and restore mechanism in case it is ever needed in regards to live data. It also creates a duplicate database that can be tested on without using a live data set. It is better to upgrade a restored instance, rather than just having a backup in the event of a required rollback.

Fix Packs and Patching

It is generally recommended to resolve and patch all issues prior to upgrading. You don't want issues to snowball as part of the upgrade process, so having issues resolved prior to beginning your upgrade is very important. If you have no active issues, updating to the latest Fix Pack or Update on both the old version of Liferay DXP and the new version is recommended. Fix Packs in Liferay DXP contain mechanisms on verifying data and ensuring data integrity, which is beneficial when preparing for an upgrade between versions.

The latest Fix Pack or Update for your previous version can be found in Liferay's Help Center.

With DXP 7.3 and 7.4, individual Fix Packs will not be released, instead, complete Update bundles will be released. For more information see the Bundle Updates section.

It is best practice to upgrade to the latest release of the target version. Upgrading to 7.4 GA1 and then moving to the latest 7.4 Update adds additional work and additional risk over going directly to the latest 7.4 Update.

License Key

Don't forget to request a new 7.4 activation key for the project. The Liferay Provisioning Team will request the following information about the environment:

- Environment hostname
- IP address
- MAC address

It is not necessary to provide all 3, and only 1 is required for Provisioning to generate a new license file.

Running the Upgrade

One of the major features introduced in Liferay DXP 7.0 was the Upgrade Tool. This continues to be available in 7.4, but with a few added benefits. Not only is the ability to run the upgrade module by module still present, but now the upgrade tool will allow you to continue an upgrade that has failed. This saves time and energy without having to restore backups every time the upgrade failed.

Upgrade Steps

1. **Set Indexing to Read Only**

   Indexing while upgrading leads to performance penalties and is an unnecessary use of system resources that could be devoted to completing the upgrade faster. To set the index to read only, an OSGi config file will have to be created.

   1. Create a file called

      ```
      com.liferay.portal.search.configuration.
      IndexStatusManagerConfiguration.config
      ```

   2. Place the file in `[Liferay Home]/osgi/config`
   3. Edit the file and add in

      ```
      indexReadOnly=”true”
      ```

2. **Find Upgrade Script**

   The upgrade tool is located in `[Liferay Home]/tools/portal-tools-db-upgrade-client` and contains all the necessary files for starting the upgrade.

3. Upgrade Related Properties

Within the upgrade tool directory, there are numerous properties files. These are similar to the portal-ext.properties files that are used when the server is running, but these are only specific to the upgrade process. The properties shown here are for use with a large database and aid in performance of the system when running the upgrade. They are not required for running the platform normally.

1. Open portal-upgrade-ext.properties
2. Insert the following properties

```
module.framework.properties.dependency.
manager.sync.timeout=500

module.framework.properties.dependency.
manager.thread.pool.enabled=false
```

3. If using Oracle database, add the following

```
hibernate.jdbc.batch_size=0
```

4. Upgrade Process Tuning Options

One of the key benefits of the upgrade tool is the ability to set its own JVM parameters and tuning. The regular day-to-day usage of the system does not reflect the same usage as the upgrade process, so they should be tuned differently.

Regular Usage of Database	Upgrade Usage of Database
95% read (SELECT)	98% write (UPDATE, DELETE, INSERT)
5% write (UPDATE, DELETE, INSERT)	2% read (SELECT)

In the upgrade tool directory, there is a `db_upgrade.sh` and `db_upgrade.bat` file. These are the files that will be used. By default, the upgrade tool has the following parameters:

```
-Dfile.encoding=UTF8 -Duser.country=US

-Duser.language=en -Duser.timezone=GMT -Xmx2048m
```

In order to change those parameters, one could modify the `.sh` or `.bat` file and run the file. The option for specifying parameters is also possible through the "`-j`" flag. An example of setting the JVM to 10GB at runtime would be:

```
db_upgrade.sh -j "-Dfile.encoding=UTF-8
-Djava.locale.providers=JRE,COMPAT,CLDR
-Djava.net.preferIPv4Stack=true -Duser.timezone=GMT -Xms8g
-Xmx8g -XX:MaxNewSize=1536m
-XX:MaxMetaspaceSize=1g -XX:MetaspaceSize=1g
-XX:NewSize=1536m -XX:SurvivorRatio=7"
```

If running on Windows, simply replace the `.sh` file extension with `.bat` or omit the extension entirely.

5. **Run Upgrade Script**

To run the upgrade, simply run the script `db_upgrade.sh` or `db_upgrade.bat` depending on the OS. On first run, the upgrade process will ask a series of questions about the type of database and the locations of various files depending on the application server type. The tool will offer default answers; you can hit "Enter" to accept these default answers without typing out the entire string.

The command line prompt will look something like this:

```
[ jboss tcserver tomcat weblogic websphere wildfly ]

Please enter your application server (tomcat):

tomcat

Please enter your application server directory (D:\
liferay\demo\liferay-dxp-7.4.13.u33\tomcat-9.0.56):

Please enter your extra library directories
in application server directory (/bin):

Please enter your global library directory in
application server directory (/lib):

Please enter your portal directory in application
server directory (/webapps/ROOT):

[ db2 mariadb mysql oracle postgresql sqlserver sybase ]

Please enter your database (mysql):

mysql

Please enter your database JDBC driver class
name (com.mysql.cj.jdbc.Driver):
```

```
Please enter your database JDBC driver
protocol (jdbc:mysql://):

Please enter your database host (localhost):

Please enter your database port (none):

Please enter your database name (lportal):

Please enter your database username:

lruser

Please enter your database password:

*******

Please enter your Liferay home
(D:\liferay\demo\liferay-dxp-7.4.13.u33):
```

6. **Post Upgrade-Script Tasks**

   a. portal-ext.properties

   Every new version of Liferay DXP comes with a few new properties settings that are defaults in the older version. It is highly recommended to review the new default settings and compare with existing settings to ensure compatibility. These settings are located inside the Liferay deployment such as `[Liferay WAR File]/WEB-INF/classes/portal-legacy-x.y.properties` where `x.y` is the current version of Liferay DXP you are on. If you have upgraded across multiple versions, you will need the older versions.

   Additionally, some `portal-ext.properties` settings may have been moved to OSGi config file settings, so certain settings will no longer be used and have been moved. For more information on config files, please refer to the documentation on System Settings.

   b. Verify Your Data

   Once the upgrade process is complete and properties have been verified, it's a good idea to start the system to make sure everything is there. If you haven't deployed any custom modules or themes, those may not show up, or in the case of the theme, it may revert to the classic Liferay theme. This may be a good time to test out core Liferay DXP functionality and ensure things are working the way you expect. A basic check would be to ensure any content is present, pages are laid out the way they should be and permissions and users are set up correctly.

Note that it may be necessary to reindex Elasticsearch post upgrade to ensure that all content displays as expected. This is required if you have moved to new Elasticsearch infrastructure, or migrated to Elasticsearch from Solr. This can be carried out from `Control Panel -> Search -> Index Actions -> Reindex all search indexes.`

7. **Deploy Modules and Themes**

   Don't forget to deploy any custom modules or themes to the upgraded system. Once they have been deployed, check again that things are working the way you expect. Don't forget that any custom modules that were built need to be at least recompiled for DXP 7.4, as noted above in the checklist.

   Other things like Liferay Marketplace Apps can also be deployed at this time as well, such as the connector for SharePoint that Liferay provides.

   Why not check everything at once? Why do two checks? Two checks allows for isolation of variables. If you run into issues, it will be easier to identify if the issue is caused by the upgrade process, an issue in core Liferay DXP 7.4 or customizations. When everything is tested in one pool, it can be difficult to isolate the root cause.

8. **Tune Application Server JVM and Other Performance Items**

   Running Liferay DXP with default JVM settings is possible, but not reflective of any real-world usage. In normal usage, a vast majority of the work is read statements on the database, with occasional write/updates; however, during the upgrade, there are few read statements and the majority of the operations are update/write. Performance tuning is dependent on many variables, such as the technology stack itself, the number of concurrent users, the number of concurrent users at peak time and the types of operations that users will be using. This process is an iterative process, usually involving changing a setting, testing that and then making further adjustments.

   Please refer to the Liferay DXP 7.4 Deployment Checklist for details on performance tuning and getting ready for going live in production.

9. **Stand up a Standalone Elasticsearch**

   In Liferay DXP 7.x, the default search engine is a stripped down, embedded version of Elasticsearch, such as Sidecar. For a production environment, it is required to replace the embedded engine with a standalone Elasticsearch setup. The bundled Elasticsearch server, including Sidecar, is not supported.

   Once the server has been setup, connect it into Liferay, deploy the necessary connectors and reindex. The system must be reindexed after an upgrade or when installing a standalone search engine.

If the system is not reindexed, Liferay DXP will not list content or users in the Control Panel interfaces, or not return any search results with known search terms.

For more information, read the documentation on installing a search engine.

10. Verify With Stakeholders

At this point, all the difficult work should be done, and it's time to verify that everything is as it should be. Talk to your various stakeholders and testers to ensure that all data and functionality is present, that the look and feel is correct and that the upgraded version meets all branding and style guidelines.

For an in-depth look at upgrading to Liferay DXP 7.4, please refer to the documentation.

Considerations for Containers

When running the upgrade in a container, there's no simple way to answer the prompts on the upgrade process. To start the upgrade process within a container, one would need to log into the container and start it there and answer the prompts. Instead of a manual start and answering prompts, the required information can be filled out in the 3 properties files found in the upgrade tool directory.

1. In `[Liferay Home]/tools/portal-tool-dx-upgrade-client`, find `app-server.properties`, `portal-upgrade-database.properties`, and `portal-upgrade-ext.properties`.
2. Fill out the necessary information in those files for the location of the app server within the container's file system, the database connection details, and `[Liferay Home]` value.
3. In `portal-ext.properties`, set `upgrade.database.auto.run=true`
4. Start the container to begin the upgrade.

Liferay Experience Cloud Self-Managed

For comprehensive documentation on upgrading the Liferay DXP instance on Liferay Experience Cloud Self-Managed, please refer to the documentation.

Post-Upgrade Considerations for 7.4

Bundle Updates

In DXP 7.4, separate Fix Packs were removed and replaced with whole bundle Updates. Updates include product fixes, like the old Fix Packs, but may also introduce new features. These features may be disabled by default to prevent disrupting existing functionality.

In a Docker or Liferay Experience Cloud Self-Managed environment, updating to a new bundle is as easy as replacing the tag in docker-compose or LCP.json. Due to the nature of how those systems work, all data is preserved, and all customizations are applied on top of those images.

Stylebooks

Introduced in 7.3, Stylebooks are a way to style and change theme elements without creating a custom theme. Stylebook elements are also variables that can be called in custom Fragments, allowing for the creation of consistent look and feel.

For more information about Stylebooks, please refer to the documentation.

Collections, Collection Pages, and Display Page Templates

First introduced in 7.2, but enhanced for 7.4, Collections, Collection Pages, and Display Page Templates are a more flexible alternative to Asset Publisher and Widget Templates. They allow for the creation of Collections, which are groupings of content like Documents and Images, which can then be dynamically displayed on a page.

For more information about Collections and Collection Pages please refer to the documentation.

Objects

Introduced in 7.4, Objects are the low-code complement to Service Builder. It allows for the creation of data fields and complex relationships between entries. If there are existing Service Builder modules and customizations, it may be possible to remove them and replace them with Liferay Objects.

For further information about Objects, please refer to the documentation.

Remote Apps

Introduced in 7.4, Remote Apps use Liferay's front-end infrastructure to register external applications with the Liferay platform and render them as widgets. If you have portlets or widgets that exist solely to render UI or information from another system, these are good candidates to replace with a Remote App.

For further information about Remote Apps, please refer to the documentation.

Moving Forward

Our Liferay Global Services team is ready to provide a deep analysis of your specific requirements, help you form a personalized upgrade plan and offer you inside knowledge on setting your company up for success with Liferay. Learn more about Liferay DXP 7.4 and the consulting services available to you by contacting sales@liferay.com or visiting liferay.com/consulting.

Liferay

220810