Upgrading DSpace

These instructions are valid for any of the following upgrade paths:

- *Upgrading ANY prior version (1.x.x, 3.x or 4.x) of DSpace to DSpace 5.x (latest version)*

For more information about specific fixes released in each 5.x version, please refer to the appropriate release notes:

- DSpace 5.x Release Notes

For more information about new features or major changes in previous releases of DSpace, please refer to following:

- Releases - Provides links to release notes for all prior releases of DSpace
- Version History - Provides detailed listing of all changes in all prior releases of DSpace

Upgrading database structure/data is now automated!

As of DSpace 5.0, the underlying DSpace database structure changes are now AUTOMATED (using FlywayDB). This means that you no longer need to manually run SQL scripts. Instead, the first time you run DSpace, it will auto-update your database structure (as needed) and migrate all your data to be compatible with the installed version of DSpace. This allows you to concentrate your upgrade efforts on customizing your site without having to worry about migrating your data!

For example, if you were running DSpace 1.4, and you wish to upgrade to DSpace 5, you can follow the simplified instructions below. As soon as you point your DSpace 5 installation against the older DSpace 1.4-compatible database, your database tables (and data) will automatically be migrated to be compatible with DSpace 5.

Automatic reindex of all content for search/browse after a database upgrade.

(Note: the upgrade to DSpace 5.6 from a prior minor version (5.0-5.5) will also trigger an update to your database schema.)

If your database structure/data is upgraded, the migration process will now trigger a reindex of all your content after deployment to Tomcat. Some repository content will be not be discoverable until this reindex process is complete. For large repository instances, this process could take some time to complete.

Optionally, you may choose to skip this automatic reindex and run a manual reindex at a later time. For more information on doing so, see the note under the optional "Upgrade your Database" step below.

Please refrain for customizing the DSpace database tables. It will complicate your next upgrade!

With the addition of our automated database upgrades, *we highly recommend AGAINST customizing the DSpace database tables/structure or backporting any features that change the DSpace tables/structure*. Doing so will often cause the automated database upgrade process to fail (and therefore will complicate your next upgrade).

If you must add features requiring new database tables/structure, we recommend creating new tables (instead of modifying existing ones), as that is usually much less disruptive to our automated database upgrade.

Test Your Upgrade Process

In order to minimize downtime, it is always recommended to first perform a DSpace upgrade using a Development or Test server. You should note any problems you may have encountered (and also how to resolve them) before attempting to upgrade your Production server. It also gives you a chance to "practice" at the upgrade. Practice makes perfect, and minimizes problems and downtime. Additionally, if you are using a version control system, such as subversion or git, to manage your locally developed features or modifications, then you can do all of your upgrades in your local version control system on your Development server and commit the changes. That way your Production server can just checkout your well tested and upgraded code.

In the notes below `[dspace]` refers to the install directory for your existing DSpace installation, and `[dspace-source]` to the source directory for DSpace 5.x. Whenever you see these path references, be sure to replace them with the actual path names on your local system.

Backup your DSpace

Before you start your upgrade, it is strongly recommended that you create a backup of your DSpace instance. Backups are easy to recover from; a botched install/upgrade is very difficult if not impossible to recover from. The DSpace specific things to backup are: configs, source code modifications, database, and assetstore. On your server that runs DSpace, you might additionally consider checking on your cron/scheduled tasks, servlet container, and database.

Make a complete backup of your system, including:

- Database: Make a snapshot/dump of the database. For the PostgreSQL database use Postgres' pg_dump command. For example:

```
pg_dump -U [database-user] -f [backup-file-location] [database-name]
```

- Assetstore: Backup the directory (`[dspace]/assetstore` by default, and any other assetstores configured in the `[dspace]/config/dspace.cfg` "assetstore.dir" and "assetstore.dir.#" settings)
- Configuration: Backup the entire directory content of `[dspace]/config`.
- Customizations: If you have custom code, such as themes, modifications, or custom scripts, you will want to back them up to a safe location.
- Statistics data: what to back up depends on what you were using before: the options are the default SOLR statistics, optional Elastic Search statistics, or the legacy statistics. Legacy stats utilizes the dspace.log files, Elastic Search stats stores data in `[dspace]/elasticsearch`, SOLR Statistics stores data in `[dspace]/solr/statistics`. A simple copy of the data directory should give you a point of recovery, should something go wrong in the update process. We can't stress this enough, your users depend on these statistics more than you realize. You need a backup.

Update Prerequisite Software (as necessary)

DSpace 5.x requires the following versions of prerequisite software:

- Java 7 (Oracle or OpenJDK)
- Apache Maven 3.0.5 or above
- Apache Ant 1.8 or above
- Database
  - PostgreSQL 9.0 or above, OR
  - Oracle 10g or above
- Tomcat 7 or above

Refer to the Prerequisite Software section of "Installing DSpace" for more details around configuring and installing these prerequisites.

Upgrade Steps

1. **Download DSpace 5.x:** Either download DSpace 5.x from DSpace.org or check it out directly from the Github repository.
   a. NOTE: If you downloaded DSpace do not unpack it on top of your existing installation. Refer to Installation Instructions, Step 3 for unpacking directives.

2. **Merge any User Interface customizations or other customizations (if needed or desired).** If you have made any local customizations to your DSpace installation they *may* need to be migrated over to the new DSpace.
   a. NOTE: If you are upgrading across many versions of DSpace at once (e.g. from 1.x.x to 5.x), you may find it easier to *first* upgrade DSpace, and *then* attempt to migrate over your various customizations. Because each major version of DSpace tends to add new configurations and features to the User Interface, older customizations may require more work to "migrate" to the latest version of DSpace. In some situations, it may even be easier to "start fresh", and just re-customize the brand new User Interface with your local color scheme, header/footer, etc.
   b. Customizations are typically housed in one of the following places:
      i. JSPUI modifications: `[dspace-source]/dspace/modules/jspui/src/main/webapp/`
      ii. XMLUI modifications: `[dspace-source]/dspace/modules/xmlui/src/main/webapp/`
      iii. Config modifications: `[dspace]/config`

3. **Edit the build.properties file (if needed)** (`[dspace-source]/build.properties`). Any settings changed in this `build.properties` file are automatically copied over to the final `dspace.cfg` file during the "Build DSpace" process (in the next step). For more information on the build.properties file, see "The build.properties Configuration Properties File" section of the Configuration Reference documentation.
4. **Build DSpace.** Run the following commands to compile DSpace :

   ```
   cd [dspace-source]/dspace/
   mvn -U clean package
   ```

   The above command will re-compile the DSpace source code and build its "installer". You will find the result in `[dspace-source]/dspace/target/dspace-installer`

   Defaults to PostgreSQL settings

   Without any extra arguments, the DSpace installation package is initialized for PostgreSQL. If you use Oracle instead, you should build the DSpace installation package as follows:
   `mvn -Ddb.name=oracle -U clean package`

Enabling and building the DSpace 5 Mirage 2 theme

Mirage 2 is a responsive theme for the XML User Interface, added as a new feature in DSpace 5. It has not yet replaced the Mirage 1 theme as the XMLUI default theme.

To enable Mirage 2, add the following to the `<themes>` section of `src/dspace/config/xmlui.xconf` , replacing the currently active theme:

```
<theme name="Mirage 2" regex=".*" path="Mirage2/" />
```

It is important to do this before executing the maven build.

Mirage 2 is not yet activated in the default "mvn package" build. To include it as part of the build, run:

```
mvn -U clean package -Dmirage2.on=true
```

The speed of this specific step of the build can be increased by installing local copies of the specific dependencies required for building Mirage 2. The Mirage 2 developer documentation provides detailed instructions for these installations. After the installation of these dependencies, you can choose to run:

```
mvn -U clean package -Dmirage2.on=true -Dmirage2.deps.included=false
```

Warning: The Mirage 2 build process should NOT be run as "root". It must be run as a non-root user. For more information see: Mirage 2 Common Build Issues

5. **Stop Tomcat (or servlet container).** Take down your servlet container.
    a. For Tomcat, use the `$CATALINA_HOME/shutdown.sh` script. (Many Unix-based installations will have a startup/shutdown script in the `/etc/init.d` or `/etc/rc.d` directories.)

6. **Update DSpace Installation.** Update the DSpace installation directory with the new code and libraries. Issue the following commands:

```
cd [dspace-source]/dspace/target/dspace-installer
ant update
```

The above command will also automatically upgrade all your existing Solr indexes (e.g. for Discovery, Statistics, OAI-PMH) to the latest version. For large instances, this may take some time. But, it is important to ensure that your indexes are usable by the latest version of DSpace.

    a. If the Solr index upgrade fails, you may need to Manually Upgrade your Solr Indexes. See the "Troubleshooting Upgrade Issues" section below.

7. **Update your DSpace Configurations.** You should review your configuration for new and changed configurations in DSpace 5.x.
    a. In the specific case of `dspace.cfg` it is recommended to start with a fresh copy of the file from the *new* version and copy your site-specific settings from the old file. Read the new file carefully to see if you need (or want) other alterations.
    b. **Please notice** that as of DSpace 4, the default search and browse support has changed from the old Lucene/DBMS-based method to Discovery.
    c. It is recommended to review all configuration changes that exist in the config directory, and its subdirectories. It is helpful to compare your current configs against a clean checkout of your current version to see what you have customized. You might then also want to compare your current configs with the configs of the version you are upgrading to. A tool that compares files in directories such as Meld or DiffMerge is useful for this purpose.
        i. Upgrading from 4.x to 5.x, notice that file config/crosswalks/google-metadata.properties uses google.citation_author instead of google.citation_authors

8. **Decide which DSpace Web Applications you want to install.** DSpace comes with a variety of web applications (in `[dspace]/webapps`), each of which provides a different "interface" to your DSpace. Which ones you install is up to you, but there are a few that we highly recommend (see below):

    a. "xmlui" = This is the XML-based User Interface, based on Apache Cocoon. It comes with a variety of out-of-the-box themes, including Mirage 1 (the default) and Mirage 2 (based on Bootstrap).*Between the "xmlui" and "jspui", you likely only need to choose one.*
    b. "jspui" = This is the JSPUI-based User Interface, which is based on Bootstrap. *Between the "xmlui" and "jspui", you likely only need to choose one.*
    c. "solr" (*required*) = This is Apache Solr web application, which is used by the "xmlui" and "jspui" (for search & browse functionality), as well as the OAI-PMH interface. **It must be installed in support of either UI.**
    d. "oai" = This is the DSpace OAI interface. It allows for metadata and bitstream (content-file) harvesting, supporting OAI-PMH (Protocol for Metadata Harvest) and OAI-ORE (Object Reuse and Exchange) protocols
    e. "rest" = This is the DSpace REST API
    f. "sword" = This is the DSpace SWORDv1 interface. More info on SWORD protocol and its usage.
    g. "swordv2" = This is the DSpace SWORDv2 interface. More info on SWORD protocol and its usage.
    h. "rdf" (*new*) = This is the DSpace RDF interface supporting Linked (Open) Data.
    i. "lni" (*deprecated*) = This is the DSpace Lightweight Networking Interface, supporting WebDAV / SOAP / RPC API. It is disabled by default as we recommend using REST or SWORD for most activities. In order to build it you must rebuild DSpace with the following flag: `mvn package -Pdspace-lni`

9. **Deploy DSpace Web Applications**. If necessary, copy the web applications from your `[dspace]/webapps` directory to the subdirectory of your servlet container (e.g. Tomcat):

```
cp -R [dspace]/webapps/* [tomcat]/webapps/
```

See the installation guide for full details.

10. **Upgrade your database** (*optional, but recommended for major upgrades*). As of DSpace 5, the DSpace code will automatically upgrade your database (from any prior version of DSpace). By default, this database upgrade occurs automatically when you restart Tomcat (or your servlet container). However, if you have a large repository or are upgrading across multiple versions of DSpace at once, you may wish to manually perform the upgrade (as it could take some time, anywhere from 5-15 minutes for large sites).
    a. First, you can optionally verify whether DSpace correctly detects the version of your DSpace database. It is **very important** that the DSpace version is detected correctly before you attempt the migration:

    ```
    [dspace]/bin/dspace database info
    # Look for a line at the bottom that says something like:
    # "Your database looks to be compatible with DSpace version ___"
    ```

    b. In some scenarios, if your database's "sequences" are outdated, inconsistent or incorrect, a database migration error may occur (in your DSpace logs). In order to AVOID this scenario, you may wish to manually run the "update-sequences.sql" script PRIOR to upgrade. This "update-sequences.sql" script will auto-correct any possible database sequence issues. In the future, we hope to automate this step to avoid any sequence problems:

    ```
    # As of 5.11, this script can instead be run via
    [dspace]/bin/dspace update-sequences

    # For 5.10 and below, you have to run this via database tools
    # General PostgreSQL example
    psql -U [database-user] -f [dspace]/etc/postgres/update-sequences.sql [database-name]

    # Example for a PostgreSQL database named "dspace", and a user account named "dspace"
    # psql -U dspace -f [dspace]/etc/postgres/update-sequences.sql dspace
    ```

    c. **WARNING:** DSpace releases PRIOR to 1.5 (i.e. 1.4.x or earlier) may need to run a single SQL statement manually to ensure your `BitstreamFormatRegistry` table does not throw errors for long mimetype strings (e.g. "Error attempting to update Bitstream Format and/or Metadata Registries org.postgresql.util.PSQLException: ERROR: value too long for type character varying(48)"). These errors may appear in your dspace logs when running the `dspace database migrate` command (see next instruction), and unfortunately may result in the 5.x data migrations (especially for EPerson metadata) to not succeed. To protect against this possible issue, run the next SQL statement on your database prior to running `dspace database migrate` (NOTE: This issue has been resolved in 6.x, as current registry updates are no longer run prior to other migrations in DS-2818)

    ```
    # Note: For 1.5 or above, this column will already be of size 256 (instead of 48)
    ALTER TABLE BitstreamFormatRegistry ALTER COLUMN mimetype TYPE VARCHAR(256);
    ```

    d. Then, you can upgrade your DSpace database to the latest version of DSpace. (NOTE: check the DSpace log, `[dspace]/log/dspace.log.[date]`, for any output from this command)

    ```
    [dspace]/bin/dspace database migrate
    ```

    e. The database migration should also *automatically trigger* your metadata/file registries to be updated (based on the config files in [dspace]/config/registries/). However, if this update was NOT triggered, you can also manually run these registry updates (they will not harm existing registry contents) as follows:

    ```
    [dspace]/bin/dspace registry-loader -metadata [dspace]/config/registries/dcterms-types.xml
    [dspace]/bin/dspace registry-loader -metadata [dspace]/config/registries/dublin-core-types.xml
    [dspace]/bin/dspace registry-loader -metadata [dspace]/config/registries/eperson-types.xml
    [dspace]/bin/dspace registry-loader -metadata [dspace]/config/registries/local-types.xml
    [dspace]/bin/dspace registry-loader -metadata [dspace]/config/registries/sword-metadata.xml
    [dspace]/bin/dspace registry-loader -metadata [dspace]/config/registries/workflow-types.xml
    ```

    f. *In DSpace 5.x, it's possible you'll encounter "Validate failed. Found differences between applied migrations and available migrations: Migration Checksum mismatch" errors during upgrades.* Please see the Database Migration Checksum Mismatch / Validate Failed known issue listed below. In this situation, it's safe to run a repair to resolve the validation failure:

    ```
    [dspace]/bin/dspace database repair
    ```

g. *If the database upgrade process fails or throws errors*, then you likely have manually customized your database structure (and/or backported later DSpace features to an older version of DSpace). In this scenario, you may need to do some manual migrations before the automatic migrations will succeed. The general process would be something like this:
    i. Revert back to your current DSpace database
    ii. Manually upgrade just your database **past** the failing migration. For example, if you are current using DSpace 1.5 and the "V1.6" migration is failing, you may need to first manually upgrade your database to 1.6 compatibility. This may involve either referencing the upgrade documentation for that older version of DSpace, or running the appropriate SQL script from under `[dspace-src]/dspace-api/src/main/resources/org/dspace/storage/rdbms/sqlmigration/`)
    iii. Then, re-run the migration process from that point forward (i.e. re-run `./dspace database migrate`)

h. More information on this new "database" command can be found in Database Utilities documentation.

By default, your site will be automatically reindexed after a database upgrade

If any database migrations are run (even during minor release upgrades), then by default DSpace will automatically reindex all content in your site. This process is run automatically in order to ensure that any database-level changes are also immediately updated within the search/browse interfaces. See the notes below under "**Restart Tomcat (servlet container)**" for more information.

However, you may choose to **skip automatic reindexing.** Some sites choose to run the reindex process manually in order to better control when /how it runs.

Skipping automatic reindexing involves two main steps, *both of which must be executed prior to restarting Tomcat (or your servlet container):*

1) First, you **must** manually update your database by running `[dspace]/bin/dspace database migrate` (See above for additional notes on this process)

2) If any database migrations are run, a empty "reindex.flag" file will be created at `[dspace]/solr/search/conf/reindex.flag` The existence of this file is what triggers the automatic reindexing when Tomcat starts up. Simply delete the "reindex.flag" file, and your site will now skip automatic reindexing. (NOTE: this file will only ever exist temporarily, and is only created when your database is modified during an upgrade or migration)

As you have disabled automatic reindexing, make sure to manually reindex your site by running `[dspace]/bin/dspace discovery -b` *(This must be run after restarting Tomcat)*

*WARNING: It is not recommended to skip automatic reindexing, unless you will **manually reindex** at a later time, or have verified that a reindex is not necessary. Forgetting to reindex your site after an upgrade may result in unexpected errors or instabilties.*

11. **Remove deprecated Database Browse Tables (optional, but recommended)**
    a. By default, DSpace now uses Discovery (backed by Solr) for its Search and Browse engine. Discovery offers additional features like filtered (or faceted) searching, and "access aware" searching which was not offered by the Legacy Search and Browse system. We highly recommend using Discovery for a better Search and Browse experience. In the future, the Legacy Search and Browse system likely will be removed.
    b. As long as you plan to use the default settings in DSpace (with Discovery enabled), you can safely remove any old Legacy browse tables (named "bi_*", where "bi" = browse index). To do so, simply run:

    ```
    [dspace]/bin/dspace index-db-browse -f -d
    ```

    c. The contents of one more leftover Legacy browse table needs to be removed, the "communities2item" table. From an SQL client, execute the following SQL (and commit the changes/purge the recyclebin if you're using Oracle):

    ```
    DELETE FROM COMMUNITIES2ITEM;
    ```

12. **Restart Tomcat (servlet container).** Now restart your servlet container (Tomcat/Jetty/Resin) and test out the upgrade.
    a. **Upgrade of database:** If you didn't manually upgrade your database in the previous step, then your database will be automatically upgraded to the latest version. This may take some time (seconds to minutes), depending on the size of your repository, etc. Check the DSpace log (`[dspace]/log/dspace.log.[date]`) for information on its status.
    b. **Reindexing of all content for search/browse:** If your database was just upgraded (either manually or automatically), all the content in your DSpace will be automatically re-indexed for searching/browsing. As the process can take some time (minutes to hours, depending on the size of your repository), it is performed in the background; meanwhile, DSpace can be used as the index is gradually filled. *But, keep in mind that not all content will be visible until the indexing process is completed.* Again, check the DSpace log (`[dspace]/log/dspace.log.[date]`) for information on its status.

        i. *If you wish to skip automatic reindexing, please see the Note above under the "Upgrade your Database" step.*
13. **Reindex SOLR Stats**. If you were previously using SOLR stats, the schema has changed with DSpace 5; you will need to reindex your stats in order to ensure all of your stats data conforms to the new schema specification. NOTE: it is safe to run a reindex on a live site, the script will store incoming usage data in a temporary core.
14. **Check your cron / Task Scheduler jobs.** In recent versions of DSpace, some of the scripts names have changed.

    a. Check the Scheduled Tasks via Cron documentation for details. Especially pay attention to the Solr Index optimization commands, which ideally should be run regularly (as noted in the previous step).

Troubleshooting Upgrade Issues

Manually Upgrading Solr Indexes

If you run into issues with the auto-upgrade of your Solr search/browse indexes (during the final part of the `ant update step`), then you may need to manually upgrade your Solr indexes. Depending on the type of failure, there are a few possible fixes.

1. If the "ant update" process failed to download the `lucene-core-3.5.0.jar`, in order to upgrade a DSpace 1.6.x, 1.7.x or 1.8.x index.
    a. You can manually download the `lucene-core-3.5.0.jar` from http://search.maven.org/remotecontent?filepath=org/apache/lucene/lucene-core/3.5.0/lucene-core-3.5.0.jar
    b. Place the lucene-core-3.5.0.jar in your `[dspace-source]/dspace/target/dspace-installer/` directory (i.e. the directory where you ran "ant update" from)
    c. Re-Run "`ant update`". This time, it should find the `lucene-core-3.5.0.jar` locally and re-attempt the upgrade of your Solr indexes.
2. If some other error occurred, you may need to **manually** upgrade your Solr indexes.
    a. **Upgrading from DSpace 1.6.x, 1.7.x or 1.8.x:** In DSpace 1.x versions, we used and older version of Solr which is no longer compatible with the current version of Solr.
        i. If you are using an older version of DSpace, you will see errors similar to this one until you manually upgrade your index:

```
Caused by: org.apache.lucene.index.IndexFormatTooOldException: Format version is not
supported (resource: segment _386q in resource ChecksumIndexInput(MMapIndexInput(path="
/space/dspace/solr/statistics/data/index/segments_37m6"))): 2.x. This version of Lucene
only supports indexes created with release 3.0 and later.
```

        ii. Manually upgrading your Solr index involves temporarily downloading an older version of Lucene (on which Solr is based), and calling its IndexUpgrader script, e.g.

```
# Download Lucene 3.5.0, which can upgrade older Solr/Lucene indexes
wget "http://search.maven.org/remotecontent?filepath=org/apache/lucene/lucene-core/3.5.0
/lucene-core-3.5.0.jar" -O lucene-core-3.5.0.jar

# Then, actually upgrade the indexes by loading the lucene-core-3.5.0.jar and calling
IndexUpgrader

# Upgrade the Usage Statistics index. Run this if you have Solr Usage Statistics enabled in
your UI.
java -cp lucene-core-3.5.0.jar org.apache.lucene.index.IndexUpgrader [dspace]/solr
/statistics/data/index/

# Upgrade the OAI-PMH indexes. Run this if you use the "oai" webapp.
java -cp lucene-core-3.5.0.jar org.apache.lucene.index.IndexUpgrader [dspace]/solr/oai/data
/index/

# NOTE: You do not need to upgrade the Discovery Search and Browse indexes as they will be
automatically rebuilt on upgrade (See previous upgrade step)
```

        iii. At this point in time, your older indexes will now be compatible with Solr / Lucene 3.5. At this point they are readable by the latest version of Solr.
        iv. However, as a final step, you should still **optimize** each of these indexes using the commands detailed in the "Upgrading from DSpace 3.x or Above" step below
    b. **Upgrading from DSpace 3.x or above:** DSpace provides optimization commands for all Solr indexes. Which ones you need to run depend on which features you are using in DSpace.

```
# First, ensure your Tomcat is started up. All of the below commands will call Solr directly,
which requires Tomcat to be running.

# Optimize Usage Statistics (based on Solr). Run this if you have Usage Statistics enabled in your
UI.
[dspace]/bin/dspace stats-util -o

# Optimize OAI-PMH indexes (based on Solr). Run this if you use the "oai" webapp.
[dspace]/bin/dspace oai import -o

# NOTE: You should not need to optimize the Discovery Search and Browse indexes, as they will be
automatically rebuilt on upgrade (See previous upgrade step)
# However, you still may wish to schedule optimizing of Discovery Search & Browse (via cron or
similar)
# [dspace]/bin/dspace index-discovery -o
```

"Property was circularly defined" errors

If, after running `ant update`, you see an error like this:

```
[dspace-src]/dspace/target/dspace-installer/build.xml:88: Property ____ was circularly defined.
```

This usually means that you are attempting to build the new version of DSpace using an outdated `build.properties` file. Specifically, the property which is reported as being "circularly defined" is likely missing from your local `[dspace-source]/build.properties` file. To resolve this issue, simply:

1. Check your local build.properties file to ensure it is up to date, specifically looking for properties which are missing and adding them in.
2. Rebuild DSpace (e.g. `mvn -U clean package`). See Upgrade Steps (step #4) above.
3. Continue the rest of the upgrade process (again from step #4) above

Database Migration Checksum Mismatch / Validate Failed

Because DSpace 5.x uses Flyway v3, during some 5.x minor upgrades you may encounter errors like this one:

```
Caused by: org.flywaydb.core.api.FlywayException: Validate failed. Found differences between applied migrations
and available migrations: Migration Checksum mismatch for migration V5.0_2014.08.08__-DS-1945-
_Helpdesk_Request_a_Copy.sql: DB=1316176070, Classpath=-872444192
```

This checksum mismatch issue is the result of a know Flyway v3 bug (whereby checksums differ if the SQL migration file has different line endings): https://github.com/flyway/flyway/issues/253

Unfortunately, based on who performed a given DSpace 5.x release, the line endings of these SQL files may be different (based on whether the person performed the release on a Windows machine or a Mac/Linux machine). This has been described in more detail in DS-3688.

In this situation, it is safe to run a "repair" on the database in order to resolve the validation failure. A "repair" simply tells Flyway to update the checksum (logged in the database) with the one calculated (from the SQL files on the Java Classpath).

```
[dspace]/bin/dspace database repair
```

NOTE: This issue ONLY exists in DSpace 5.x releases, as it only affects Flyway v3. As of DSpace 6.0, we've upgraded to Flyway v4, which fixes this checksum validation bug.