

# Multi-Version Upgrade Tips and Tricks

This page provides some simple tips/tricks you can use to upgrade DSpace over multiple versions (e.g. 1.5.x -> 1.6.x -> 1.7.x -> 1.8.x -> 3.x -> 4.x). In the notes below `[dspace]` refers to the install directory for your existing DSpace installation, and `[dspace-source]` to the source directory for DSpace 4.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 ANY 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.

## Upgrading Across Multiple Versions

Depending on your local use cases and needs, there are two primary options for upgrading across multiple versions:

### Option 1: Perform each major upgrade in succession

With this option, you are just performing each major upgrade one by one following the upgrade instructions. At times, it may be a bit more difficult, but the advantage is that it is more likely to bring forward any customizations you have made locally (e.g. in branding your UI or similar).

As an example, if you wanted to upgrade **from DSpace 1.5.x to DSpace 4.x**, you'd perform **each** of these upgrades, one-by-one, in order:

- [Upgrading From 1.5.x to 1.6.x](#)
- [Upgrading From 1.6.x to 1.7.x](#)
- [Upgrading From 1.7.x to 1.8.x](#)
- [Upgrading From 1.8.x to 3.x](#)
- [Upgrading From 3.x to 4.x](#)

Likely, you'd want to perform some local testing *between each upgrade* to ensure that each upgrade is successful. Otherwise, if there are later issues, you may not be able to easily tell which upgrade caused the problems.

It is also recommended to read the Release Notes for each major version.

### Option 2: Install Fresh and Upgrade your Data

The second option is to actually install a fresh installation of the latest version of DSpace, and then simply upgrade your data (primarily the database). This is a much easier multi-version upgrade process, however it does **not** automatically upgrade all of your configurations or customizations (as you'll be migrating to a fresh, uncustomized installation). So, depending on how complex (or important) your customizations were, this may not fully meet your needs.

1. Install the version of DSpace you wish to upgrade to by following the latest installation instructions:
  - a. For example: to upgrade to DSpace 4.x follow the instructions at [Installing DSpace](#).
  - b. At this point, you'll have a fresh copy of DSpace, but no data
2. Make a backup copy of your old DSpace database. You'll use this to upgrade your data to the latest version of DSpace:
  - a. For example, in PostgreSQL use the "pg\_dump" command to backup all your old data

```
pg_dump -U [database-user] -f my-dspace-db-backup.sql [database-name]
```

3. Create an exact replica of your old DSpace database (this is mostly just for safety purposes – you don't want to lose any of your Production data!)
  - a. For example, in PostgreSQL use the "createdb" and "psql" commands:

```
# Create a replica database. I've called this one "dspace-upgrade" just as an example
createdb -U [database-user] -E UNICODE dspace-upgrade

# Load the old data into this replica database
psql -U [database-user] -f my-dspace-db-backup.sql dspace-upgrade
```

4. Now, upgrade this replica database to be compatible with the version of DSpace you wish to upgrade to. This is done utilizing the "database\_schema-\*.sql" upgrade scripts provided in the [dspace-source]/dspace/etc/postgres/ folder.
  - a. For example, suppose you are upgrading from DSpace 1.5.x to 4.x. That means you'll be upgrading your database from being 1.5.x compatible to being 4.x compatible. In order to do that, you will run **all these database upgrade scripts in order** (again this example is just for PostgreSQL):

```
# Upgrade 'dspace-upgrade' database from 1.5.x to 1.6.x compatibility
psql -U [database-user] -f [dspace-source]/dspace/etc/postgres/database_schema_15-16.sql dspace-upgrade

# Upgrade 'dspace-upgrade' database from 1.6.x to 1.7.x compatibility
psql -U [database-user] -f [dspace-source]/dspace/etc/postgres/database_schema_16-17.sql dspace-upgrade

# Upgrade 'dspace-upgrade' database from 1.7.x to 1.8.x compatibility
psql -U [database-user] -f [dspace-source]/dspace/etc/postgres/database_schema_17-18.sql dspace-upgrade

# Upgrade 'dspace-upgrade' database from 1.8.x to 3.x compatibility
psql -U [database-user] -f [dspace-source]/dspace/etc/postgres/database_schema_18-3.sql dspace-upgrade

# Upgrade 'dspace-upgrade' database from 3.x to 4.x compatibility
psql -U [database-user] -f [dspace-source]/dspace/etc/postgres/database_schema_3-4.sql dspace-upgrade
```

5. You now have a 4.x compatible database (in "dspace-upgrade")! So, you can dump it's data out to a file:
  - a. For example, to dump "dspace-upgrade" to a file named "my-dspace-upgraded-db.sql":

```
pg_dump -U [database-user] -f my-dspace-upgraded-db.sql dspace-upgrade
```

6. Remember your "fresh" installation of the newer version of DSpace (in this case DSpace 4.x)? Well, now you will just move your old data over to that fresh installation.
  - a. First, you need to move over your upgraded Database data. That's done by removing the default "fresh installation" database, and replacing it with your upgraded version:

```
# Delete the default (MAKE SURE IT IS EMPTY) "fresh install" Database on the NEW DSpace
dropdb -U [database-user] [database-name]

# Recreate an empty DB
createdb -U [database-user] -E UNICODE [database-name]

# Load in the newly upgrade data (from "my-dspace-upgraded-db.sql")
psql -U [database-user] -f my-dspace-upgraded-db.sql [database-name]
```

- b. Next, copy over your old DSpace "Assetstore" directory into the newly installed version of DSpace. This assetstore directory contains all the actual content files

```
cp -R [path-to-old-dspace]/assetstore/* [path-to-new-dspace]/assetstore/
```

- c. Finally, reindex all of your DSpace content in the newly installed version of DSpace. For example, assuming you are using Discovery faceted/filtered search/browse (the default in DSpace 4.x), you'd run:

```
[dspace]/bin/dspace index-discovery -f
```

7. At this point, you should have a fresh installation of a newer version of DSpace with your content migrated to it. As a final step, you may wish to re-configure or re-customize the fresh installation (based on your old settings).

- a. Also be sure to review all the Upgrade Instructions for the versions you "skipped over". Sometimes there are important notes or details in there!