

# Upgrading DSpace

---

[hilton.gibson@gmail.com](mailto:hilton.gibson@gmail.com)

# Introduction

How do you manage changes so that they do not affect the uptimes of the production system too much and are controlled by all interested parties?

Use a change management system!

[http://en.wikipedia.org/wiki/Change\\_Management\\_\(ITSM\)](http://en.wikipedia.org/wiki/Change_Management_(ITSM))

# Change Management (1)

Build a test/development system on another server, typically an old retired server. At Stellenbosch University library we built: <http://repository.sun.ac.za> (Only available on-campus). This is the machine which you use to test changes and upgrades of software before you implement it on the production server.

Try not to make major changes to the system during peak usage times of the year. Pick a time when there are very few users on the system.

If you have service level agreements with the IT department, then inform them of changes that affect them at least two weeks ahead of the time.

## Change Management (2)

When you want to implement a change or upgrade you will need to inform submitters, reviewers and metadata editors. For this, setup a mailing list and ensure all submitters, reviewers and metadata editors join the list. For normal users, send a notification to the campus communications manager and then when the time arrives for the change, put the system into maintenance mode.

Before changing anything on the production server ensure your backups are working. It is a good idea to keep incremental backups for a period of at least seven days.

# Schedule of upgrades

Upgrading involves three systems.

1. The repository software each year.
2. The server software every 3 years.
3. The hardware every 4 years, if not virtualised or in the cloud.

# DSpace Upgrade - Version Strategy

Stellenbosch University Library tries to stay one major version behind the current DSpace version because we do not have expert java programmers to fix bugs. So we stay super stable by staying one version behind the latest version. We let those with expert programming resources fix bugs in the latest version while we enjoy the stable benefits of the previous version.

# DSpace Upgrade - Release Notes

Before upgrading read the release notes in detail. Below are links to the release notes of supported versions;

[Versions 5.X](#)

[Versions 4.X](#)

[Versions 3.X](#)

# DSpace Upgrade - General Guidelines

Using an Ubuntu desktop, the program "meld" is installed. "meld" allows you to compare two folders for differences. To do a comparison, the working version is copied to my desktop and the new version is extracted to my desktop as well. Using meld I investigate all the differences and update the new version with new settings as appropriate. When I am satisfied that the new version has most of my customisation from the old version, then I test on my development server. I do more debugging on the development server as needed and then finally activate it on the production server and test. On the production server we test for a couple of days.

# DSpace Upgrade - Reference Architecture

Upgrades of DSpace could be made a lot simpler if a single reference architecture were adopted.

See the link below for more details;

[http://wiki.lib.sun.ac.za/index.php/SUNScholar/Reference\\_Architecture](http://wiki.lib.sun.ac.za/index.php/SUNScholar/Reference_Architecture)

Until then DSpace software upgrades are going to be complex and difficult.

# DSpace Upgrade - Essential Preparation

1. Backup Databases - PostgreSQL and SOLR
2. Backup folders with previous customisations
3. Perform the upgrade.

See links below;

[http://wiki.lib.sun.ac.za/index.php/SUNScholar/Upgrading/DSpace/General\\_Notes#Step\\_1\\_-\\_Backup\\_Databases](http://wiki.lib.sun.ac.za/index.php/SUNScholar/Upgrading/DSpace/General_Notes#Step_1_-_Backup_Databases)

[http://wiki.lib.sun.ac.za/index.php/SUNScholar/Upgrading/DSpace/General\\_Notes#Step\\_2\\_-\\_Backup\\_folders\\_with\\_customisations](http://wiki.lib.sun.ac.za/index.php/SUNScholar/Upgrading/DSpace/General_Notes#Step_2_-_Backup_folders_with_customisations)

# DSpace Upgrade - Methodology Guidelines

Using an Ubuntu desktop, the program "meld" is installed. "meld" allows you to compare two folders for differences.

To do a comparison, the working version is copied to my desktop and the new version is extracted to my desktop as well.

Using meld I investigate all the differences and update the new version with new settings as appropriate.

When I am satisfied that the new version has most of my customisation from the old version, then I test on my development server.

I do more debugging on the development server as needed and then finally activate it on the production server and test.

# DSpace Upgrade - Checklist

Below is a basic checklist of items to do during the upgrade;

1. Main database upgrade - With DSpace versions => 5.X the main database is automatically upgraded.
2. The SOLR database is also automatically upgrade with DSpace versions => 5.X
3. Check for changes to the language “messages.xml” file.
4. Check for changes to the discovery browse and search indexes file - discovery.xml.
5. Check for changes to the handle server.
6. Check for changes to the daily tasks.

# Server Upgrade

Since we choose to use the Ubuntu server as our server operating system, we follow the [LTS release](#) upgrade path.

The major server software that is upgraded is as follows:

Tomcat Java webapp server

Java runtime software

PostgreSQL Database

# Server Upgrade - Methodology

A test system is built using the same software versions as the present production system.

On the test system a server operating system upgrade is done.

A check is done to see what changed and what needs special upgrade attention.

When everything is working well on the test system then the upgrade is done on the production system with the changes that the test system identified.

# Server Upgrade - Procedure

Login using SSH to the server and then execute the following;

1. `sudo apt-get install update-manager-core`
2. `sudo do-release-upgrade -c`
3. `sudo do-release-upgrade -m server`

Go find a coffee...

# Server Upgrade - Tips

If the upgrade fails in the beginning, then remove any non-standard sources in the `/etc/apt/sources.list` file and the `/etc/apt/sources.list.d/` folder and try again.

When asked whether to keep the maintainer's version of a config file or keep your own custom version of a config file, always select the maintainer's version.

Make a note of these files and then after the upgrade go back and check the config files for any customisation you would like to apply as you did to the old server version.

Take careful note of the PostgreSQL upgrade notes prompted on the command line.

# Server Upgrade - PostgreSQL Database

Take careful note of the PostgreSQL database upgrade. See the link below;

[http://wiki.lib.sun.ac.za/index.php/SUNScholar/Upgrading/Server\\_Software/Step\\_2](http://wiki.lib.sun.ac.za/index.php/SUNScholar/Upgrading/Server_Software/Step_2)

Make a backup of the database before doing any upgrade.

**!!You have been warned!!**

# Hardware Upgrade - Part 1

This is the very tricky one, if you are running your server on a bare metal machine. Think about asking for expert linux help to do the upgrade. If your server is virtualised then this procedure is not needed.

To avoid bare metal upgrades in the future, it is highly advised that you virtualise your server. Ask central campus IT to help you with this.

However there is always a performance price when you virtualise. In addition, since DSpace is a java webapp, you should realise, java is itself a virtual machine (JVM), so if you virtualise the hardware, then you are virtualizing, an already, virtual machine!

## Hardware Upgrade - Part 2

For this reason, bare metal machines always have the best performance for java webapps.

However, performance is not an issue if you have a small repository and very few people connected at one time as collection submitters/managers.

If you anticipate having a big repository then you should consider going bare metal.

*Bare metal is preferred for performance reasons.*

# Hardware Upgrade - Methodology

1. Prep new machine
2. Export and import
3. Enable Tomcat webapps on new machine
4. Enable "handle server" on new machine
5. Swap hostnames
6. Precautionary Principle

**Keep the old machine running for a while until the new machine is stabilised. This way you have a live backup in case of big problems.**

# Conclusion

To recap, upgrading involves three systems.

1. The repository software each year.
2. The server software every 3 years.
3. The hardware every 4 years, if not virtualised or in the cloud.

<http://wiki.lib.sun.ac.za/index.php/SUNScholar/Upgrading>