

# BackupRestore

needsupdate

## Backing Up and Restoring a DSpace Instance

This is a strawman guide to backing up your DSpace instance. Comments, additions, errata etc. invited.

TODO: Good restore info

When it comes to backing up DSpace there are a few broad backup strategies you can take:

### File system backup

Back up DSpace and the applications and data files as they are, directly from the file system. The advantage to this is that, if you have a compatible hardware/OS stack, it is very straightforward to restore a DSpace backup to a functioning instance. The disadvantage is that if you need to restore to a different platform (or maybe even a similar platform with a newer OS version) your backup will not be easy to restore as many of the restored apps won't work.

At an absolute minimum, you should back up:

- [dSPACE-source] if you've made any customisations
- [dSPACE]/assetstore
- [dSPACE]/config
- [dSPACE]/history and [dSPACE]/log if you want to preserve activity records
- PostgreSQL, especially /usr/local/pgsql/data (or wherever the data dir is)

Really you could carry on down the tool and O/S stack; Java, Tomcat it's a judgement call as to how far down you go.

### Storage layer backup

Just back up the data; i.e. the RDBMS tables and [dSPACE]/assetstore (in the default config). You assume that you will be able to recreate a software/hardware environment that will run DSpace, and then you will just need to restore the data.

The advantage of this approach is that you only have to back up the data; as long as you can reconstruct the correct DSpace and PostgreSQL versions, it doesn't matter if the underlying hardware and OS are different. i.e. you're not so reliant on having an exact replica of the hardware and OS around to restore the backup.

However, you DO need the right version of DSpace (including any customisations you may have made) since the underlying DB schema can change between DSpace versions (usually when the second number has changed, e.g. 1.1.1 -> 1.2).

You also need a compatible version of PostgreSQL to restore the backup. There are two approaches here. One is a 'file system level backup', where you simply backup the relevant PostgreSQL data files from the system (e.g. /usr/local/pgsql/data), as [described here](#). Alternatively (and preferably) you can do an 'SQL dump' of the database, in which you get PostgreSQL to dump out the contents of the DSpace database as an SQL file, which you can feed back into another PostgreSQL instance to recreate the database. This is [described here](#).

In short, what you'd do here is:

- back up your assetstore.dir(s), /dSPACE/assetstore in the default config
- as part of your backup process, get PostgreSQL to write a dump of the contents of the dSPACE database, and back up that dump
- back up any customisations you've made to the DSpace code
- maybe back up any config files (dSPACE.cfg etc)

Some notes in <http://dspace.org/technology/system-docs/storage.html> about backing up and restoring your Postgres database and the bitstream store.

### Use import/export tools

#### DSpace AIP Backup & Restore Process

In DSpace 1.7.0 and above, there is now an AIP (Archival Information Package) Backup and Restore process available. For more information see: [AIP Backup and Restore](#)

Back up the data using the batch item exporter. In other words, you'd use the batch item exporter to export all of the contents of your DSpace to location X, and back X up.

From the point of view of being able to restore a functioning DSpace system to how it was before a failure, this isn't a complete solution, since the item exporter does not export all the information. Specifically, the following information is not present:

- e-person records
- authorisation policies
- the community and collection structure
- primary bitstreams (i.e. for multi-file HTML items)
- Dublin Core type registry (though the exported items will have all the qualified Dublin Core metadata in so this would be reasonably simple to restore)
- the bitstream format registry (the format of bitstreams would have to be re-determined on re-import)

There are a few other issues right now:

- there are some issues with a batch export -> batch import round trip
- the batch export tool does not currently have an option to export everything in the system, so you'd need to have a separate export for every collection or item, which would require extra scripting.
- you need to make sure you also backup the 'map' files so you still have the Handles for each item

The experimental METS exporter tool does export richer information about each item, but there's no corresponding importer yet! The CWSpace project at MIT is looking at providing this in the next few months though.

Although there are issues, this strategy does have benefits in terms of long-term preservation. First and foremost, you don't need a functioning DSpace instance to be able to access your backed up data. If you have another application or converter that can understand the export format, you can put the backed up data into a different system if you wish. The exported files are also independent of any particular version of PostgreSQL or other tools, and hopefully also independent of DSpace version; the batch import/export format should remain backwards-compatible.

## Notes

For the short term, strategy 1 or 2 (or a mix) is probably the best way to go; if you have the backup capacity, following strategy 3 in parallel has the potential for long-term benefits.

As requested, a short description of the procedure we follow.

- When installing the server, a partition is created for the system and another system for data. Every partition is then divided in Logical Volumes. This enables restore of only required logical volumes.
- We try to install software as much as possible in the system partition, and data on the data partition. Log files however, tend to end up in the system partition.
- DSpace and the PostgreSQL databases are put on the data partition.
- On the data partition there are separate logical volume for backup, downloads, etc.
- After installation of the server, a full (Ghost) image is taken.
- A daily hotbackup is made of the PostgreSQL database via a crontab. This can be done while PostgreSQL is running. We use the following parameters (pg\_dumpall -o -c -v), but check documentation for the PostgreSQL version you are using. These daily backups are kept for a month.
- A weekly incremental backup of all partition is made to disk (and tape). For the incremental backup the PostgreSQL server has to be shut down, else the PostgreSQL database files are not included in the backup.
- During the weekly backups we leave Tomcat running and replace the DSpace homepage with a maintenance page. This is not a very good solution, because most users, crawlers and OAI-harvesters do not start at the DSpace homepage. This generates many Internal Error Messages. We do not use Apache and are still looking for a good solution in Tomcat.

## Other methods

Please check out: [http://wiki.lib.sun.ac.za/index.php/SUNScholar/Disaster\\_Recovery](http://wiki.lib.sun.ac.za/index.php/SUNScholar/Disaster_Recovery) for a method on Ubuntu servers.