

Installing DSpace 1.7 on Ubuntu

One of the limiting factors to the adoption of DSpace by the community has historically been the the difficulty in installing DSpace and all of its prerequisites. This goal of this guide is to provide a simplistic path that will get a user up-and-running as quickly as possible.

The simple act of installing DSpace is actually quite simple, as changing between DSpace versions (eg between 1.6 and 1.7) can be done in a few minutes, however is it the installation and configuration of the prerequisites that is usually time consuming and difficult. Within Ubuntu, this is standard stuff, so we can have Ubuntu do all of that dirty work for us. The following is steps that are performed from Terminal, the command line interface in Ubuntu.

This how-to was originally written for Ubuntu 10.10 (Maverick Meerkat) and has been tested with Ubuntu 11.04 (Natty Narwhal).

- [Install Prerequisites](#)
 - [Make sure your sources are up to date before beginning](#)
 - [Install the server stack of Tomcat \(web server\) and PostgreSQL \(database\)](#)
 - [Install the Compile / Build tools](#)
- [Configure the Prerequisite Software](#)
 - [Create the database user \(dspace\)](#)
 - [Allow the database user \(dspace\) to connect to the database](#)
 - [Create the dspace database](#)
 - [Configure Tomcat to know about the DSpace webapps.](#)
- [Download and Install DSpace](#)
 - [Create the \[dspace\] directory.](#)
 - [Download the Source Release](#)
 - [Compile and Build DSpace](#)
 - [Fix Tomcat permissions, and restart the Tomcat server](#)
- [Test it out in your browser](#)
- [\(OPTIONAL\) Change to using Sun/Oracle Java JDK](#)
 - [Enable the Canonical Partners repository.](#)
 - [Install Sun Java](#)
 - [Change the in-use Java to Sun Java, as opposed to OpenJDK](#)

Install Prerequisites

Make sure your sources are up to date before beginning

```
sudo apt-get update
```

Install the server stack of Tomcat (web server) and PostgreSQL (database)

```
sudo apt-get install tasksel
sudo tasksel
```

- Select the following packages

```
[*] LAMP server
[*] PostgreSQL database
[*] Tomcat Java server
```

Install the Compile / Build tools

```
sudo apt-get install ant maven2
```

Configure the Prerequisite Software

Create the database user (dspace)

```
sudo su postgres
createuser -U postgres -d -A -P dspace
exit
```

Allow the database user (dspace) to connect to the database

```
sudo vi /etc/postgresql/8.4/main/pg_hba.conf
# Add this line to the configuration: local all dspace md5
sudo service postgresql restart
```

Create the dspace database

```
createdb -U dspace -E UNICODE dspace
```

Configure Tomcat to know about the DSpace webapps.

```
sudo vi /etc/tomcat6/server.xml
# Insert the following chunk of text just above the closing </Host>

<!-- Define a new context path for all DSpace web apps -->
<Context path="/xmlui" docBase="/dspace/webapps/xmlui" allowLinking="true"/>
<Context path="/sword" docBase="/dspace/webapps/sword" allowLinking="true"/>
<Context path="/oai" docBase="/dspace/webapps/oai" allowLinking="true"/>
<Context path="/jspui" docBase="/dspace/webapps/jspui" allowLinking="true"/>
<Context path="/lni" docBase="/dspace/webapps/lni" allowLinking="true"/>
<Context path="/solr" docBase="/dspace/webapps/solr" allowLinking="true"/>
```

Download and Install DSpace

Create the [dspace] directory.

The [dspace] directory is where the running dspace code will reside.

```
sudo mkdir /dspace
```

Download the Source Release

The source release allows you to customize every aspect of DSpace. This step downloads the compressed archive from SourceForge, and unpacks it in your current directory. The dspace-1.x.x-src-release directory is typically referred to as [dspace-src].

```
wget http://sourceforge.net/projects/dspace/files/DSpace%20Stable/1.7.2/dspace-1.7.2-src-release.tar.bz2
tar -xvjf dspace-1.7.2-src-release.tar.bz2
```

Compile and Build DSpace

The source release that has been obtained is human readable source code, and must be compiled to machine code for the server to run it. "mvn package" compiles the source code, and "ant" will do all the work necessary to initialize the database with the DSpace schema, and copy all of the compiled machine code to a location where the web server can serve it.

ant fresh_install will populate the dspace database and [dspace] directory with new information. This will overwrite any existing installation of DSpace that you may have.

For upgrades the better command to use would be ant update, as it doesn't alter the database or modify your assetstore.

```
cd dspace-1.7.2-src-release
mvn -U package
cd dspace/target/dspace-1.7.2-build.dir
sudo ant fresh_install
```

Fix Tomcat permissions, and restart the Tomcat server

This guide follows the convention where the tomcat user will own all of the files in [dspace], so we have to change the owner of the files to tomcat6. Restarting tomcat will deploy the dspace webapps that are now ready to be viewed.

```
sudo chown tomcat6:tomcat6 /dspace -R
sudo service tomcat6 restart
```

Test it out in your browser

That is all that is required to install DSpace on Ubuntu. There are two main webapps that provide a similar turn-key repository interface

<http://localhost:8080/xmlui>

<http://localhost:8080/jspui>

(OPTIONAL) Change to using Sun/Oracle Java JDK

An optional step that can be done after installation would be to switch to the Sun/Oracle Java JDK. The taskel task to install Tomcat installs the default OpenJDK which is a viable form of Java, however the official recommendation of DSpace is to use the Sun/Oracle Java JDK which offers better performance and other proprietary enhancements.

Note: Recent Change to Java Distribution



Oracle has changed their partner distribution license, which means that Canonical/Ubuntu has had to purge Sun-Java-6 from their repository. Thus, updating Ubuntu will remove Sun-Java from your system, and installing Sun-Java from the partner repository will no longer work.

See more: <https://lists.ubuntu.com/archives/ubuntu-security-announce/2011-December/001528.html>

Enable the Canonical Partners repository.

The Sun Java is available in the partners repository which makes for an easy installation. From the GUI this can be changed by going to Software Sources. Change "natty" to the name of the Ubuntu version you're using (if you aren't using Ubuntu 11.04)

```
sudo vi /etc/apt/sources.list
# Uncomment the line: deb http://archive.canonical.com/ubuntu natty partner
sudo apt-get update
```

Install Sun Java

```
sudo apt-get install sun-java6-jdk sun-java6-plugin
```

Change the in-use Java to Sun Java, as opposed to OpenJDK

First we list the available jdk's installed on the system, then we set the sun java to be the new default.

```
sudo update-java-alternatives -l
# java-6-openjdk 1061 /usr/lib/jvm/java-6-openjdk
# java-6-sun 63 /usr/lib/jvm/java-6-sun
sudo update-java-alternatives -s java-6-sun
```

Tomcat ignores the system default and continues to use OpenJDK. To make Tomcat use Sun Java instead, add the following line to /etc/default/tomcat6:

```
JAVA_HOME=/usr/lib/jvm/java-6-sun
```

You will need to restart Tomcat to make it pick up the change (sudo service tomcat6 restart).