# DSpace IDE - Eclipse, Git, Maven, and Tomcat



# Introduction / Goals / Objectives

This wiki page is targeted for those individuals who work with DSpace but have need to explore or modify the DSpace source code and would like to setup their DSpace installation(s) within an Integrated Development Environment so as to have access to the features and tools that those tools provide.

# Information on Prerequisite Tools

## SCM Tools

The SCM you use is certainly up to you (and your organization). However, Duraspace provides source code repositories for DSpace via Git.

### Git

The DSpace Git repository is located here: https://github.com/DSpace/DSpace.git

## Build Tools

Utilities necessary for deploying an instance of DSpace

### Maven

The Apache Maven project is a software development tool targeted primarily toward dependency management but also helps to keep code and projects well organized. For more information on Maven please check out the Apache Maven website. This is a required build tool for working with DSpace source code (and is not required for the pre-built versions that can be downloaded). If you're reading this tutorial and are wanting to setup an IDE for working with DSpace, this tool will be a prerequisite.

### Ant

Apache Ant is a Java library and command-line tool. It's purpose is to drive processes that are described in build files (xml format). These build files contain "targets" that can be run, which can also be dependent on each other, in turn creating process chains. Ant is another requisite of building DSpace from source.

## Servlet Containers

A servlet container. There are many (Apache Tomcat, Apache Geronimo, GlassFish, and Jetty, to name a few). Though this section may be expanded later to Geronimo and/or Jetty, I'll be covering Apache Tomcat to start.

### Tomcat

Apache Tomcat is an open source software implementation of the Java Servlet and JavaServer Pages technologies. This is application that will run the servlets created through the DSpace deployment process.

### Geronimo

placeholder

### Jetty

placeholder

## Eclipse IDE

This wiki page concerns setting up Eclipse, but there are others (which are more popular among DSpace developers). If you like, please browse to the pages on NetBeans (oracle site, dspace wiki) or IntelliJ (jetbrains site, dspace wiki).

The current version of Eclipse (as of 2014-02-13) is 4.3, codenamed "Kepler". This is the version that will be the subject of this wiki page. Keep in mind that there are several very popular Eclipse "packages". The significance of an Eclipse package is which tools or plugins are bundled with that particular download. The three most popular (in ascending order of size / number of included tools/plugins) are:

1. Eclipse Classic - This version is only the base Eclipse software (3rd most popular).
    a. This version includes no additional tools or plugins except for Git Support.
2. Eclipse for Java Developers - This version includes some basic plugins, here are the highlights:
    a. Git
    b. m2e
    c. Mylyn
3. Eclipse for Java EE Developers - This version includes more, but not all that the regular Java version provides.
    a. Mylyn
    b. DataTools
    c. JDT (Java Development Tools)
    d. JST (J2EE Standard Tools)
    e. WST (Web Standard Tools - this encompasses JDT, JST, and some others)
    f. RSE (Remote System Explorer - otherwise known as Target Management)

I highly recommend choosing Eclipse for Java EE Developers as it is very likely that you make good use of all the included tools save perhaps for Mylyn (depending on your organization / institution's development environment). Also, this is the package that I'll be using for this tutorial (other plugins will be added via the Eclipse update tool, the Eclipse Marketplace or the Eclipse "plugins" folder). Feel free to make notes within this page for different versions of installation instructions.

## Eclipse Plugins

### SCM Connectors

Both of these plugins connect Eclipse with Subversion repositories. Their strengths and weaknesses vary a little depending on the task. I tend to go with Subclipse primarily due to this article from person.

#### Egit

#### Subclipse

#### Subversive

### m2e

Maven to Eclipse is included in some of the packages. However it is not working with the current Release of DSpace 4.0.

### m2e Extensions

These are extensions for the m2e plugin and are available through the m2e marketplace (accessible from within Eclipse).

#### m2e-scm

#### m2e-cvs

#### m2e-subclipse

#### m2e-subversive

#### m2e-egit

#### Embedded Maven 3 (optional)

An installation of Maven 3 available from within the Eclipse IDE. Using an embedded version of Maven will limit your memory space to that allowed to Eclipse, for this reason I generally recommend against using this option and instead installing Maven externally from Eclipse (somewhere on your filesystem).

### Sysdeo Tomcat Connector

This an Eclipse plugin which will allow for starting/stopping/restarting tomcat from within Eclipse and will also enable debugging of DSpace source code from within Eclipse.

### oXygen XML Editor

XMLUI development

This plugin won't be fantastically useful unless you plan to be doing some development within the XMLUI interface driven by Apache Cocoon. If you will be doing work in Cocoon or on themes for DSpace, this is (in my experience) a very valuable tool.

# Installation / System Requirements

# Configuring Your Environment

## Installing Git

Eclipse Kepler comes with Git support and you could clone the repository with Eclipse . However this HOW-TO will not explain the cloning of the repository with Eclipse because it doesn't work as expected.
So you have to install Git on your operating system from:

- Git Homepage (includes info on downloading & installing on major platforms)

Using a Local Repository instead?

If your institution plans to instead use a local repository (mercurial, git, svn...) for your local development, you can do so (you will just need to download the DSpace Source Code and import into your local repository).

## Installing Maven

For most Linux distributions, you should be able to just install the Maven client available in your distribution's repositories.
For all other operating systems, you can install the latest version from the Apache Maven site.

## Checkout DSpace from GitHub

Need a tutorial on Git/GitHub?

If you need help/tips/resources on DSpace development with Git/GitHub, or just tutorials on Git in general, you may want to check out our DSpace Development with Git page.

- ***IMPORTANT NOTE:*** If you plan to do a larger amount of DSpace development or local changes, you may wish to first "fork" the DSpace GitHub Repository (https://github.com/DSpace/DSpace) to your own GitHub account. This will create your own copy of the DSpace source code under your GitHub account (e.g. https://github.com/\[your-username\]/DSpace). You can then checkout your own forked repository to work from and commit local changes to (push changes to). For more information, see the GitHub help page on "Forking a Repo".

To work on DSpace you have to make a clone of the remote repository on your local machine. To do this:

- move into the directory where you want to have the DSpace source folder

- clone the repository with:

```
git clone https://github.com/DSpace/DSpace.git
```

***IMPORTANT NOTE:*** With the command below you will work on the **master branch** of DSpace to change this you can either change the branch you are working on after the cloning or you can clone the remote repository with the **additional option -b <branch-name>**
For example:

```
git clone https://github.com/DSpace/DSpace.git -b dspace-4_x
```

(for existing branches see https://github.com/DSpace/DSpace.git)

## Build DSpace and create Eclipse Configuration Files

To execute the following commands you need to have the DSpace sources downloaded or cloned on your hard drive and Maven installed.

`YOUR_DSPACE_SOURCE_FOLDER = The complete path to the dspace sources. Something like /home/myusername/workspace /DSpace`

```
cd DSpace # enter the folder with the dspace sources
mvn package # Build DSpace with maven
mvn -Declipse.workspace=YOUR_DSPACE_SOURCE_FOLDER eclipse:configure-workspace # Create DSpace workspace with
the maven eclipse plugin
mvn eclipse:eclipse # Create Eclipse config files with the maven eclipse plugin
```

## Import Projects into Eclipse

If you have done the steps described before you can import the maven modules into Eclipse with this steps:

- Open Eclipse
- Choose: File -> Import -> General -> Existing Projects Into Workspace
- Click Next
- Click Browse
- You don't have to select anything just click ok

That's it.you should see multiple Projects now. To make a Maven overlay just use the projects without dspace- in their names (like project xmlui).

If you want to have Git support in Eclipse you might have to:

- Select all projects right click and select Team -> Share Project...
- Choose Git and click next
- Just press Finsh and you are done

## Debugging with Tomcat

For debugging with Eclipse and Tomcat you need to have Tomcat installed, the webapps deployed (see installation HOW-TO) and activate remote debugging (see: http://wiki.apache.org/tomcat/FAQ/Developing#Q2)

1. Open tomcat settings in (on Ubuntu) /etc/default/tomcat7
2. Uncomment line:
    a. `JAVA_OPTS="${JAVA_OPTS} -Xdebug -Xrunjdwp:transport=dt_socket,address=8000,server=y,suspend=n`
3. Restart tomcat like: sudo service tomcat7 restart
4. In Eclipse choose a module/project like dspace-api click on Debug Configurations... -> Remote Java Applications
5. Enter your tomcat address (like localhost) and the Port 8000.
6. At tab source add the other modules (like dspace-xmlui)

To actually debug the code you have to connect to tomcat by clicking on the new created Debug Configuration (in our case its called dspace-api) and visit the DSpace site with your browser.

## Build and Install DSpace

Initially and after every change to the code you have to build the project with maven and deploy the changes to Tomcat (install/update DSpace). For further information see Installing DSpace and Rebuild DSpace

## A Quick Feature Run-Through

## Tips & Hints

Working with Multiple DSpace Instances

Working with Multiple Tomcat Instances

Working with Multiple Solr Indexes