IDE Integration - DSpace and IDEA

- **Authors**: James Rutherford, Richard Jones, Stuart Lewis, Tim Donohue
- **Version**: 1 February 2011

Table of Contents:

Instruction Video for Version v13 on MacOS

Installing IDEA

This guide covers version 11.1 of IDEA, which you can get from here: http://www.jetbrains.com/idea/. Instructions for installing & running can be found on the website and in the readme file included with the download.

Creating a DSpace project from a GitHub clone

- From the main window, click "VCS", "Check out from Version Control", then "GitHub".
- When it prompts you for a repository, click the "+" icon and enter the URL for the DSpace Repository in GitHub which you wish to work with (e.g. https://github.com/DSpace/DSpace.git OR if you have your own fork: https://github.com/\[your-username\]/DSpace.git).
    - ***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".
- Next, select the "Remote Branch" you wish to develop on. A few hints: [[Copied from NetBeans, actually it is different for IntelliJ IDEA]]
    - Branches named "dspace-#_#_x" (e.g. dspace-1_8_x) are Bug Fix / Maintenance branches. So, the latest code in that bug-fix or maintenance release will be available on that branch. This code tends to be more stable overall. **As such, we recommend most developers use the appropriate Bug Fix / Maintenance branch for their local development.**
    - The branch named **master** is roughly equivalent to the old SVN Trunk. As such, it may not be as stable, but it includes the latest & greatest code which is being prepared for the next major release. *Unless you know what you are doing, we do NOT recommend running this code in Production. It is essentially "unsupported" until it is officially released.*
    - If you wish to work from a "tagged" (official release) version of DSpace (e.g. 1.8.2), you can download those releases as Tarballs/Zips from: https://github.com/DSpace/DSpace/tags You could then use that Tarball/Zip to import it into your own Git/GitHub or SVN repository as you see fit.

Once the source has been cloned, you will be asked whether you want to create a project (click "Yes").

- Select "Import project from external model", then "Maven".
- All the options on the subsequent page should be correct (make sure 'Exhaustive search' is ticked, it might not be), so just click "next", and then select 'all', and the 'postgres' (or 'oracle') profile and all the projects.
- Then all you have to do is choose a name for the project and click "finish".

Note: If you haven't already defined a JDK, you may not be able to do so at this point, despite being given the option. Instead, create the project without a JDK, then later open the Settings dialog, then go `Project Settings -> JDKs` and add your JDK here

Git/GitHub Hints & Tips

The following are a few hints/tips which you may want to utilize to ease your development processes with IDEA and GitHub:

1. **Fork your own Repo to store your local changes:** As recommended above, you really should think about forking your own copy of the DSpace GitHub repository. As GitHub describes in their "Fork a Repo" guide, forking lets you create your own personal copy of the codebase. It not only provides you a place to put your local customizations. It also provides an easier way to contribute your work back to the DSpace community (via a GitHub Pull Request).

2. **For easier Fetch/Merge, setup an "upstream" repository location:** This is only really relevant if you have your own personal "fork" (see #1). If you have forked the DSpace GitHub repository, then you may want to setup an "upstream" remote that points at the central DSpace GitHub repository. This is described in more detail in the GitHub "Fork a Repo" guide. Perform the following:
    - On the command-line, change directory to your local machines' cloned DSpace git repository, which is also your DSpace source directory (e.g. `cd [dspace-src]`)
    - Run the following 'git' command from that directory:

      ```
      git remote add upstream git://github.com/DSpace/DSpace.git
      ```

      (Technically you can name it something other than "upstream". But, "upstream" is just the GitHub recommended naming convention).
    - For more information about how this comes in handy, see the section below on "#Fetch & Merge Example via IDEA".

Fetch & Merge Example via IDEA

blocked URL	This assumes you've followed the #Git/GitHub Hints & Tips listed above, and have forked your own personal copy of DSpace's GitHub as well as setup an "upstream" remote link. **This is just one example of how you can perform these tasks.**

1. **Fetch changes from DSpace Main GitHub:** You fetch (and later merge) changes that have occurred in the central DSpace GitHub Repository:
    - *From IDEA*: Right-click on the "DSpace Parent Project" (root project) and then select: *Git -> Repository -> Fetch*. This will pop up a window that will allow you to easily select the "upstream" configured repository to fetch the latest changes from, and allow you to choose the "master" branch to apply them to. Once you click "Finish", a new "upstream/master" branch will be created locally with the latest changes to be merged.
    - *From Command-line* in your DSpace source directory (e.g. `cd [dspace-src]`): `git fetch upstream`
2. **Merge changes into your Local Git Repo:** Remember, "fetching" changes just brings them into your local-machine's copy of the Git repository. You'll then need to merge those changes with yours and push the changes back to your personal public GitHub repository.
    - *From IDEA*: Right-click on the "DSpace Parent Project" (root project) and then select: *Git -> Repository -> Merge Changes*. This will pop up a window to let you select which "branch" to merge into your currently checked out code. If press "Select", you'll see a new branch called "upstream/master" under "Branches -> Remote". Selecting that branch will merge the latest code from "upstream/master" into whatever branch you currently are working with (e.g. "master").
    - *From Command-line* in that same directory: `git merge upstream/master`
3. **Quick Status of Local Git Repo:** If you want to see what happened, you can look at the "Status" information:
    - *From IDEA*: Right-click on the "DSpace Parent Project" (root project) and then select: *Git -> Show History*. Click the "Search" button (without entering any search info). It will bring back results that will show you where the HEAD pointer is (latest commit in your local machine's git repo) versus where the 'origin/master' is (latest commit in your personal GitHub repo).
    - *From Command-line* in that same directory: `git status` (will tell you how many "commits ahead" of 'origin/master' you now are)
4. **Push Merged Code up to your Personal GitHub Repo:** Finally, assuming all went well, you can push your changes back up to GitHub into your public personal repository:
    - *From IDEA*: Right-click on the "DSpace Parent Project" (root project) and then select: *Git -> Repository -> Push*. This will pop up a window that will allow you to select the "origin" repository (your personal fork in GitHub), and allow you to choose the "master" branch to push to.
    - *From Command-line* in that same directory: `git push origin master`

Building and installing DSpace for the first time

Go to the command line, and go to the root of your new project. Run:

```
mvn package
```

These steps might take a while to download all the dependencies, but that should only happen once.

Note: you may notice, if you look in IDEA, that at this stage it will detect a number of web facets. Ignore this for the time being, we will come back to it. They will be available under the flashing cog on the bottom right of the screen for future reference.

After this completes:

- in the `dspace` directory navigate to `target/dspace-<version>.dir/` (the version may vary depending on which part of the svn tree you are using) and edit the `config/dspace.cfg` file according to your local requirements (this is the file that will be deployed into your `<dspace>` directory).
- make sure you have sufficient permissions on your `<dspace>` directory and that the correct database exists, etc. This is all covered in the install docs, but this differs in one important respect: your user will need to have write permissions on the `<dspace>` directory. This is because when you run the ant tasks from IDEA they will run with your permissions, and `ant update` will fail if you can't write to `<dspace>`.
- run `ant fresh_install` from this directory. You can do this in IDEA by right clicking `build.xml`, and selecting `Add as Ant Build File`. When the build file dialog opens, select `fresh_install` and hit the run button at the top of the panel.

One-click maven builds

- Go to "Run" -> "Edit configurations", then click the "+" icon (in top left) and select Maven.

Name

- In the name field at the top enter something useful like "mvn clean package" to describe the run task

Parameters Tab

- For the "Working directory", you want `<dspace-src>/dspace/`.
- Under "Goals", input "clean package"
- In Profiles, put either "postgres" or "oracle", depending on your setup.
- In the "Before launch" section, uncheck "make". Everything else should also be unchecked

General Tab

- If necessary, check "override" for the Maven Home Directory option and enter the path to your maven install directory (e.g. `/usr/share/maven-bin-2.0`)
- If necessary, check "override" for the Maven User Settings Directory option and enter the path to your settings file (something like `~/.m2/settings.xml`)

Runner Tab

- Put the path to your "live" config in the properties section at the bottom (the property key is "`dspace.config`", and the value is the absolute path to your `dspace.cfg`). The `ant fresh_install` task from the previous section should have pushed this file into your `<dspace>`directory.
    - For each config, click "Add..", and then add the proper name and value. Minimally, you should specify `dspace.config` property.
- Check the "skip tests" box (advisable, unless you want to run all Unit Tests every time you build)

Once you have this configured, click "Apply" at the bottom and then click "OK".

You should also run it immediately, because the previous build will have the incorrect `dspace.cfg` built into it. You can do this using the quick launch from the IDEA tool bar: in the center of the toolbar at the top is a pull-down menu which contains all your pre-configured run tasks. Select the one you have named "mvn clean package", and hit the green arrow to the right of it. You should see maven building your project in a window at the bottom of IDEA, and the content will be similar to when you ran it from the command line earlier.

Web Facets

At some point during the above installation process, IDEA may notify you of new web facets detected. These correspond to the individual WAR files and thus web applications which we are working on. If you do not confirm the detection of these facets straight away, they will be available on the bottom left of the screen by clicking on the flashing gear/cog.

Each of these web facets represents a web application provided by DSpace. This will include, for example, the LNI, the JSP-UI, the XML-UI and the OAI interface.

In the dialog, select "Accept". If you ignored the dialog, they should be available under the flashing 'cog' icon at the bottom of your IDEA window.

Deploying DSpace

The next step in getting fully integrated with this IDE is to set it up with ant and tomcat.

Ant Integration

- First, set up Ant by clicking the "Ant build" icon on the far right margin of the window.
- If you used Ant to build DSpace above, then your previous configuration will remain. Otherwise, add a new one by clicking the "+" icon in the upper left.
- Here, you need to select the `build.xml` file from `<dspace-src>/dspace/target/dspace-<version>.dir`.
- If you can't select or find the build.xml file, you may need to tell IDEA to stop ignoring/excluding the _target_ folder. Go to File > Project Structure, select Modules, DSpace, and then click the red X next to "target" under the heading "Excluded Folders"
- Now, in either case, right click on the top-level of the tree and select "properties" and click "add". In the boxes provided, enter "config" and the path to your live `dspace.cfg`. Click "OK".

Tomcat Integration

Note

Tomcat integration is only possible if you have IDEA Ultimate edition. The free community version does not offer this feature. If you are an official DSpace developer, you can contact one of the other Committer's to learn about how you can get ultimate edition.

For Tomcat integration, I chose to download and install tomcat 6 in `/opt` to keep it separate from the 5.5 release that I had previously installed with my package manager. There are a few reasons for this, but mostly it's because we need to assign pretty weak privileges to the tomcat directory because we will be deploying the webapp as a normal user (run something along the lines of `chmod -R jim:tomcat /opt/apache-tomcat-6.0.14 ; chmod -R g+rwx /opt/apache-tomcat-6.0.14` as root).

Once you have ant and tomcat ready, you can create a new build configuration for these two.

- Go to "Run" -> "Edit Configurations", click the "+" in upper left and select "Tomcat server" -> "Local".

Name

- Supply a sensible name for the run task, such as "Tomcat"

Server Tab

- Define the application server here; if it is not available in the pull-down menu you can configure it using "Configure"
- Uncheck "start browser"; it's a total nuisance having the tomcat target start a new browser every time you run it
- Uncheck "make"

Deployment Tab

- Select the web facets previously detected to deploy
- You may need to click on the "+" and select "Artifact"
- From there, select the facet name (e.g. `jspui:war`, `xmlui:war`)
    - **Note**: there may be other web facets available, such as `dspace-jspui-webapp:war`; these can be deployed but are not connected to the dependency libraries so unless you have tomcat set up specifically to use shared dependencies you will get tomcat errors on trying to load these pages; always use the simple named facet (e.g. `jspui:war`) to deploy.
    - You will not be able to deploy any applications which are listed as "Web (dspace-parent)"; this is OK, you just need to deploy the others.
- When you are finished click "OK"

Once you have this set up, you can deploy DSpace, by selecting the tomcat deployment from the menu bar, and hitting the green play button next to it. Hit the play button on the opened dialog, and IDEA will deploy the applications at the context paths specified in this section.

DSpace 6 (Services) and debugging command line

Because I wanted to debug command-line main

Remove: <scope>provided</scope> from servlet-api in many pom.xml files.

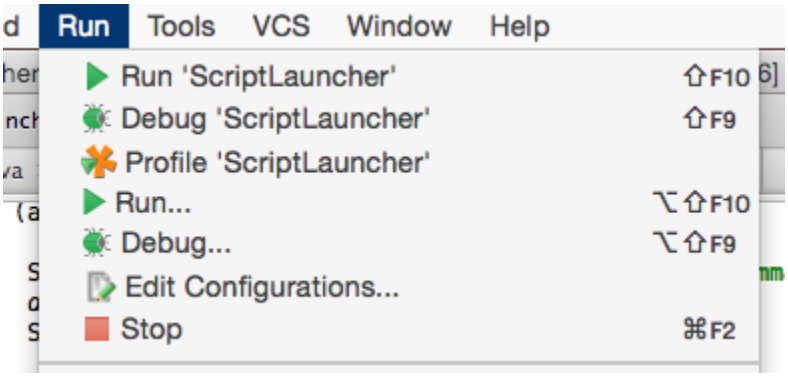
```
<dependency>
   <groupId>javax.servlet</groupId>
   <artifactId>servlet-api</artifactId>
</dependency>
```

Add some dependencies to:

dspace-services/src/main/java/org/dspace/services/caching/CachingServiceImpl.java

```
import javax.servlet.ServletRequest;
import javax.servlet.http.HttpSession;
```

Navigate to dspace-api/src/main/java/org/dspace/app/launcher/ScriptLauncher.java, and click Run --> Edit Configurations...



And fill in the Configuration of:

Main class: org.dspace.app.launcher.ScriptLauncher

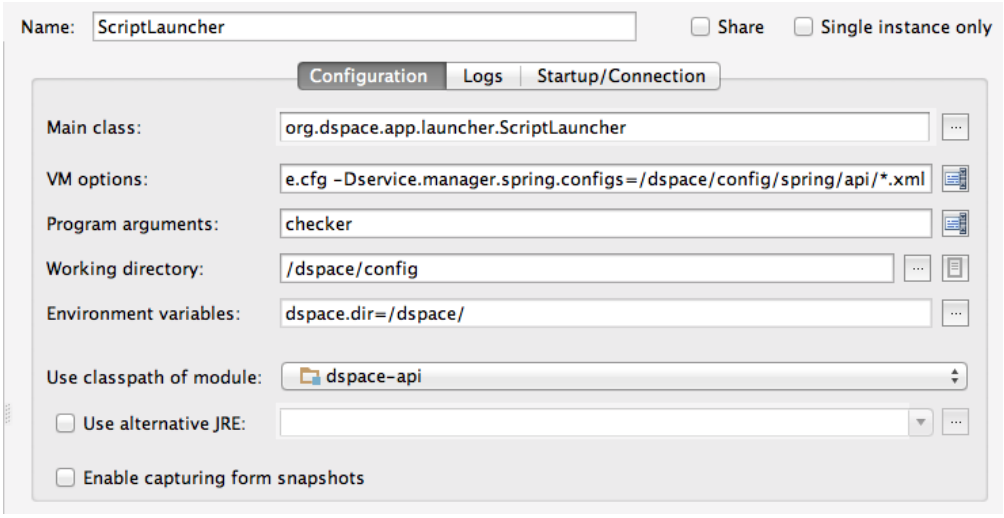
VM options: -Ddspace.dir=/dspace -Ddspace.configuration=/dspace/config/dspace.cfg -Dservice.manager.spring.configs=/dspace/config/spring/api/*.xml

(I was debugging the checksum checker, so I passed the args that would go to /dspace/bin/dspace)

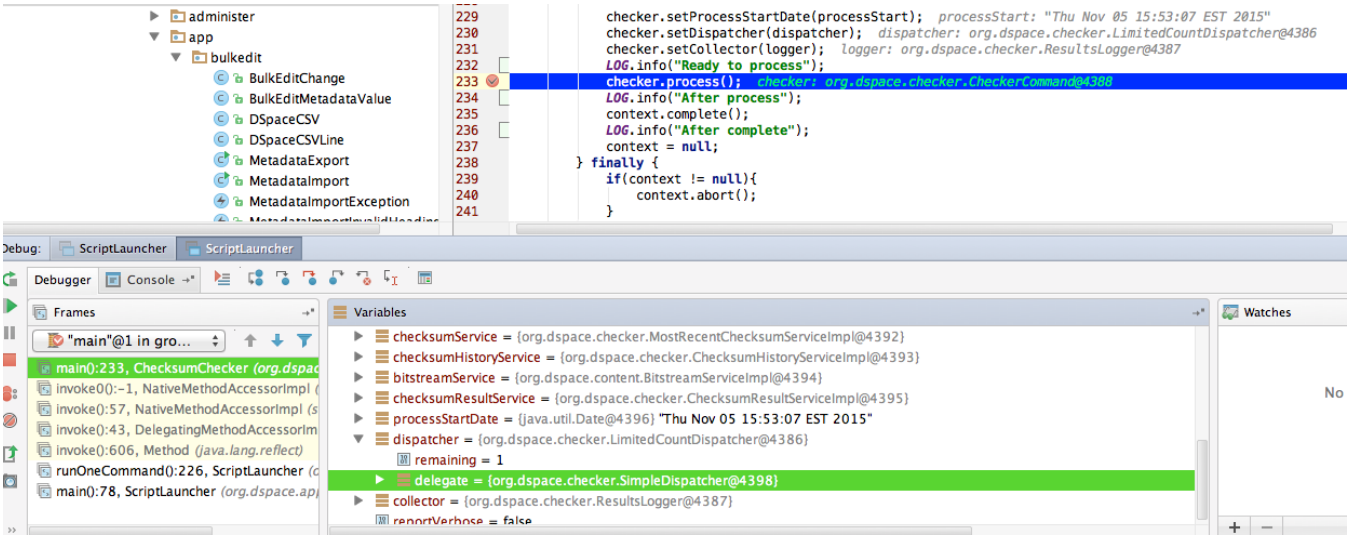
Program arugments: checker

Working directory: /dpsace/config

Environment variables: dspace.dir=/dspace/



From then, you can click Run -> Debug "ScriptLauncher", or click the green bug button, and the debugger should start up.



Tips

If you leave the "run" dialog boxes open at the bottom, you have a quick way to re-run the build and deployment tasks. The maven build can be re-run by clicking the green icon that looks like ">>", and likewise for the ant + tomcat deployment, except you'll have to click "stop" first, and wait for tomcat to shut down.

If you see an error that indicates that the dspace.dir folder cannot be found in your Tomcat/bin folder, (such as *java.io.FileNotFoundException: /opt/local /share/java/tomcat6/bin/dspace.dir/config/dspace.cfg (No such file or directory)*) when you attempt to call the Tomcat runconfig, you can try adding a symlink in your Tomcat/bin folder, like so:

```
    cd /opt/local/share/java/tomcat6/bin/
sudo ln -s /dspace/ '${dspace.dir}'
```

It's not pretty, but it'll get the job done.

Useful plugins

IDEA has loads of plugins available to install. A couple that may be useful are:

- SQL Query Plugin
- regexPlugin
- Jetty Integration
- IntelliTail