

## Contribute to the netcdf-java library

Robin,

I use Eclipse Mars.1 (4.5.1) (Java EE) with the Buildship plugin installed from the update site:

<http://download.eclipse.org/buildship/updates/e45/releases/1.0>

In Eclipse, thredds is affected by this Gradle bug: <https://issues.gradle.org/browse/GRADLE-2327> Workaround in Eclipse is to find Windows / Preferences / Java / Compiler / Building / Circular dependencies and change it to Warning.

Once Buildship is installed and the compiler settings are adjusted, import the thredds Gradle projects with File / Import / Gradle / Gradle Project and select the top-level thredds directory. Eclipse should create an Eclipse project for each Gradle module.

Once Buildship has finished configuring and building the new projects, you can run the tests for a module in Eclipse by right-clicking on src/test/java and selecting Run As or Debug As then JUnit Test. You can also run tests in packages or individual test classes.

Kind regards, Ben.

On 19/04/16 03:57, Christian Ward-Garrison wrote:

Hi Robin,

Here [\[1\]](#) is how to build and publish the library. Testing is tricky because many of our tests rely on a large collection of datasets that is not (yet) publicly available. However, I think we've almost completely isolated those to the cdm-test and it modules. Testing other modules should work fine. For example, if you're working on something in the cdm module, run:

```
./gradlew --info :cdm:test
```

to run only the tests in cdm. Unfortunately, I cannot provide you any specific instruction about Eclipse (we use IntelliJ IDEA here), but I'd be very surprised if Eclipse doesn't have a way to run tests in a Gradle-built project.

Cheers, Christian

[\[1\] http://www.unidata.ucar.edu/software/thredds/current/netcdf-java/tutorial/SourceCodeBuild.html](http://www.unidata.ucar.edu/software/thredds/current/netcdf-java/tutorial/SourceCodeBuild.html)

On Mon, Apr 18, 2016 at 6:57 AM, Péricé Robin <perice.robin@?> wrote:

Hello,

Is there any tutorial or documentation on how to build the library and run the JUnit tests on Eclipse ?

Regards,

Robin

[netcdf-java mailing list netcdf-java@?](mailto:netcdf-java mailing list netcdf-java@?) For list information or to unsubscribe, visit: [http://www.unidata.ucar.edu/mailling\\_lists/](http://www.unidata.ucar.edu/mailling_lists/)

[netcdf-java mailing list netcdf-java@?](mailto:netcdf-java mailing list netcdf-java@?) For list information or to unsubscribe, visit: [http://www.unidata.ucar.edu/mailling\\_lists/](http://www.unidata.ucar.edu/mailling_lists/)

-- Ben Caradoc-Davies <ben@?> Director Transient Software Limited <<http://transient.nz/>> New Zealand

[netcdf-java mailing list netcdf-java@?](mailto:netcdf-java mailing list netcdf-java@?) For list information or to unsubscribe, visit: [http://www.unidata.ucar.edu/mailling\\_lists/](http://www.unidata.ucar.edu/mailling_lists/)

## For developers using THREDDS github

De: John Caron <caron@?> Fecha: 30 de abril de 2015 17:23:33 GMT+02:00 Para: NetCDF-Java community <netcdf-java@?>, THREDDS community <thredds@?> Asunto: [netcdf-java] For developers using THREDDS github

Hi all:

If you build TDS/NetCDF-java from source, you may have noticed that many branches disappeared from the thredds [GitHub?](#) repository yesterday. The most notables changes are the removal of the 4.6.0 branch and the renaming of the 4.6.1 branch to master.

The rationale behind these changes is to simplify the development process, both for the main developers and for external contributions through github. Our previous model involved creating new branches for each version under development, such as 4.6.1. Once a new version was released, that branch was considered frozen, and the new version branch was created. This process of continually switching branches creates problems with easily merging community contributed pull requests (among other problems).

We have now moved to a model where the branch `master` denotes the current active development branch. This is the branch to use for any contributions to THREDDS. The code for all previous releases is still easily accessible using the tags created for each release. For instance, the code for 4.6.0 can be checked out from the `v4.6.0` tag:

```
git checkout -b 4.6.0 v4.6.0
```

If you had code changes on 4.6.0, you'll need to rebase them on master:

```
git rebase master your_branch_name
```

If you were working based on 4.6.1, you just need to point the existing branch at master:

```
git branch -u origin master
```

So, in general, you want to be working with the source code on the master branch. (There may be exceptions where you need to be on an older or newer branch, please let us know if you think that's the case).

We apologize for any inconvenience these changes have caused. Please let us know if you see any problems.

The THREDDS development team