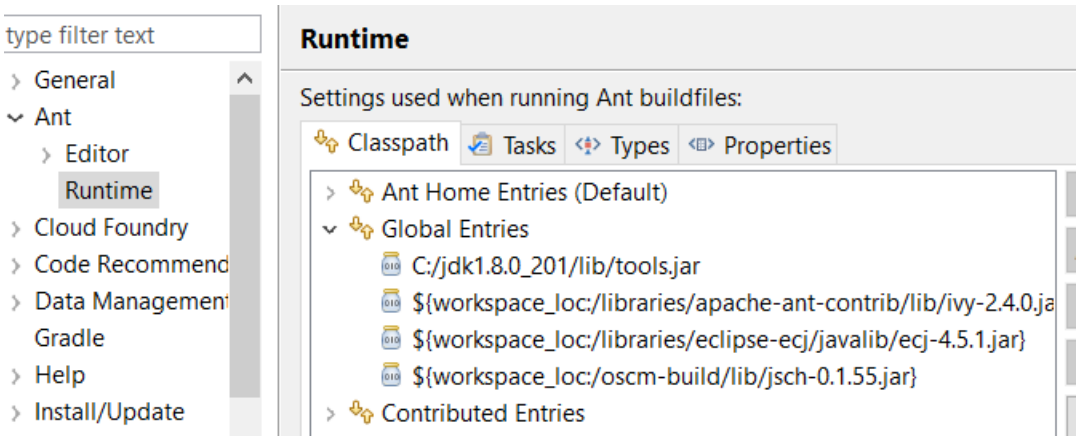
## Building OSCM in eclipse

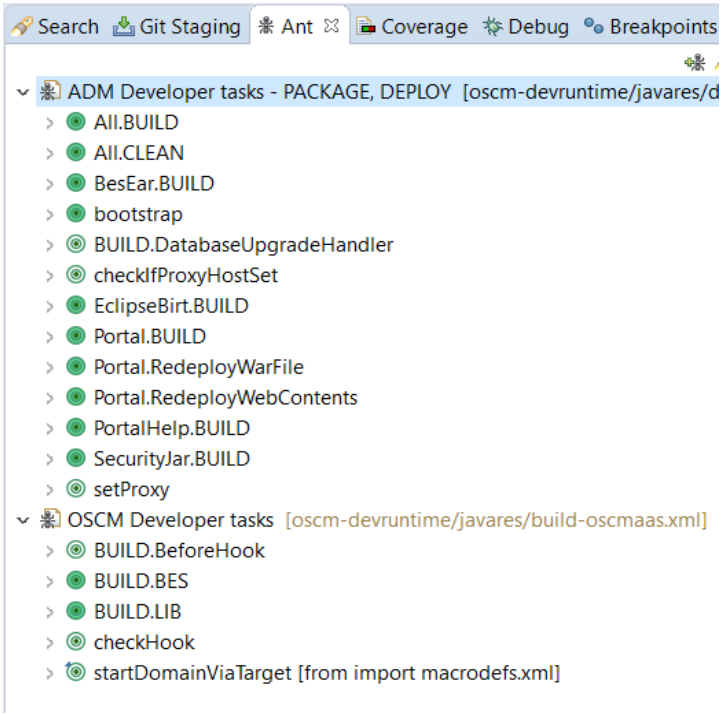
This is a quick instruction for developers to build and deploy OSCM core artifacts in eclipse.

Start your eclipse an perform following steps.

1. From the Package Explorer chose **Import > General > Existing Projects** into Workspace and select the oscm root folder from the cloned repo.
2. In Window > Preferences > Validation check Suspend all Validators and tick off all other options
3. In Window > Preferences > **ANT > Runtime** add **tools.jar** from JDK 1.8, ivy, ecj and jsch from the workspace to the ANT classpath

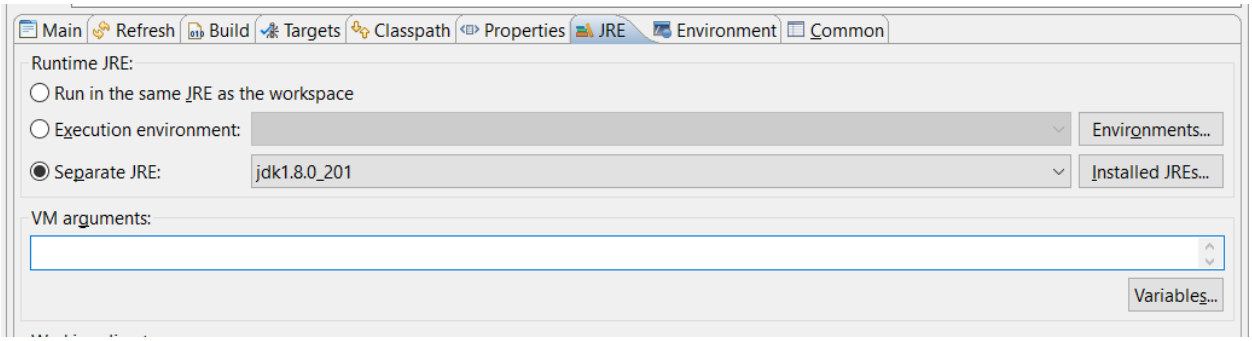



4. Open the **ANT view** in eclipse. Locate the following files in your workspace and add them to the ANT View:
   - Open /oscm-devruntime/javares/build-oscmaas.xml
   - Open /oscm-devruntime/javares/devscripts/build-dev-PackageDeploy.xml

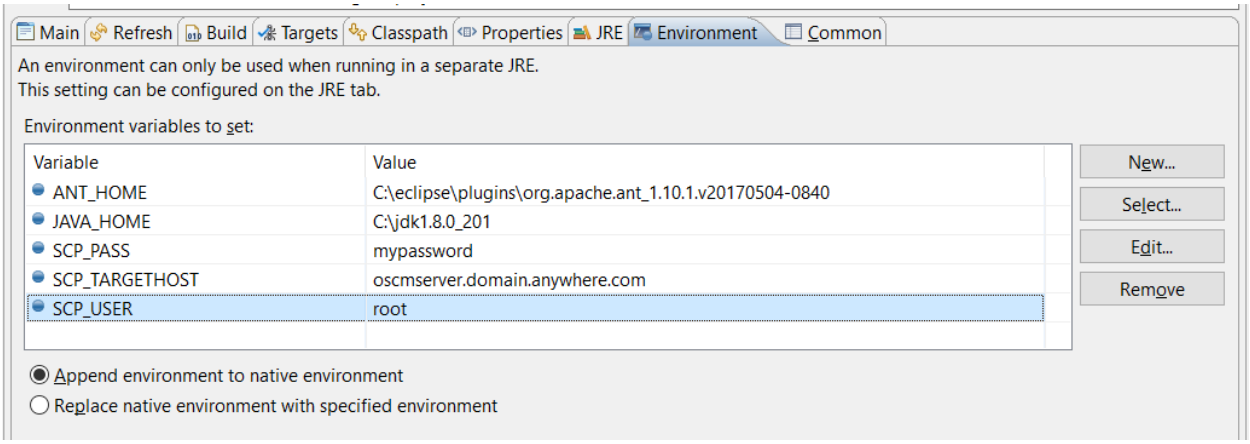


5. Right click on the ANT file oscm-devruntime\javares\build-oscmaas.xml and select **Run As > 2 Ant Build…**

6. Make sure you run ANT as separate Java process (on same JDK)




7. Expose ANT_HOME and JAVA_HOME for the process.




Additionally you can expose SCP settings. This is required for deployment targets (ANT file mentioned in 4.2). Enter the location and SCP credentials for connecting your OSCM server host.

8. Click Apply an Close.
9. Run the targets BUILD.LIB and BUILD.BES from oscm-devruntime\javares\build-oscmaas.xml

## Redeploying OSCM Portal Artifacts

Perform the following steps to exchange oscm-portal.war or your modified JSF pages on a running OSCM test server.

1. Right click on the ANT file /oscm-devruntime/javares/devscripts/build-dev-PackageDeploy.xml and select **Run As > 2 Ant Build…**
2. Repeat step 6 to 8 of above and provide SCP settings for connecting your OSCM test server: SCP_TARGETHOST, SCP_USER, and SCP_PASS.
3. Use the targets **Portal.RedeployWarFile** to exchange the war file, or **Portal.RedeployWebContents** for deploying your web content modifications.

   **Note:** The latter is the fastest way to patch JSF pages, since it even does not require to build. It exchanges web contents, like xhtml files, directly into the deployed application. Make sure that your Java and remaining source is always in sync with the deployed application!

Have fun!