Wiki »

Developer Setup

Extra libraries needed to install
- libgfortran

Setup Binary Distribution of uFrame
1. Inside your base home directory, make a uframes directory: `mkdir uframes`
2. Change to that directory: `cd uframes`
3. Download the uframe binary install from [http://spcm.omaha.us.ray.com/software/ooi/uframe_ooi_20140908.tar.gz](http://spcm.omaha.us.ray.com/software/ooi/uframe_ooi_20140908.tar.gz)
   - Or if you have the nas mounted `/race_nas/ooi/software/uframe_ooi_20140908.tar.gz`
4. Extract the binary uframe to that directory: `tar -xvf /spcmshare/software/ooi/uframe-ooi-phase2-2014_08_07.tar.gz`
5. Edit your `.bashrc` to include the uFrame Java on your path.

```
export UFRAME1_HOME=/home/${USER}/uframes/ooi
export JAVA_HOME=${UFRAME1_HOME}/uframe-1.0/java
export PATH=${JAVA_HOME}/bin:${PATH}
```

Setup Eclipse IDE
1. Change to your home directory: `cd`
2. Extract the latest Eclipse from the NAS: `tar -xvf /race_nas/software/eclipse/eclipse-standard-luna-R-linux-gtk-x86_64.tar.gz`
3. Rename the folder: `mv eclipse eclipse-standard-luna-R-linux-gtk-x86_64`
4. Extract the legacy Eclipse from the old NAS: `tar -xvf /spcmshare/software/eclipse3.8_linux64.tar.gz`
   - [http://spcm.omaha.us.ray.com/software/eclipse3.8_linux64.tar.gz](http://spcm.omaha.us.ray.com/software/eclipse3.8_linux64.tar.gz) works as well if you don't have /spcmshare mounted
5. Rename the folder: `mv eclipse eclipse3.8_linux64`

Setup gerrit ssh key
- Create key
  1. In Eclipse open Window->Preferences and navigate to General->Network Connections->SSH2.
  2. Open Key Management tab and click Generate RSA Key...
  3. Optionally fill out passphrase. (Without passphrase private key is not secure.) NOTE: This passphrase will be required in Setup Repo.
  4. Copy public key contents from text box.
  5. Select "Save Private Key."
- Add key to gerrit
  1. Ask CM to setup gerrit account for you.
  3. Log in as your gerrit user.
  4. Click settings in the top right corner of gerrit web interface.
  5. Click SSH Public Keys on the left.
  6. Click Add Key and paste in public key text (copied previously from Eclipse or copied from `~/.ssh/id_rsa.pub`).
  7. Click Add.
- Change permissions on key file generated by eclipse
  1. In a terminal cd to `~/.ssh`
  2. Run `chmod 600 id_rsa` (chmod your private key to read/write for your user only)
  3. [Restart Eclipse](#)
- **NOTE:** log out and log back in after adding ssh key to gerrit to make sure your system knows to use the new key

Setup git user configuration
These settings need to match your name and email in Gerrit exactly (case sensitive)
1. set username

2. set email

```
git config --global user.email "your_email@example.com"
```

**Setup Repository**

- Run `git clone ssh://ooicm.omaha.us.ray.com:29418/ooi.git`
  
  Initialized empty Git repository in `/home/thahn/git/ooi/.git/`
  remote: Counting objects: 2137, done
  remote: Finding sources: 100% (2137/2137)
  remote: Total 2137 (delta 766), reused 2125 (delta 766)
  Receiving objects: 100% (2137/2137), 29.02 MiB | 24.31 MiB/s, done.
  Resolving deltas: 100% (766/766), done.

  - Change directory to the ooi repo just cloned: `cd ooi`
  - Update all of the git submodules: `git submodule update --init`
  - download the commit message git hook from gerrit

```
scp -P 29418 ooicm.omaha.us.ray.com:hooks/commit-msg .git/hooks/
```

**Submodule registration**

- Submodule `submodules/AWIPS2_baseline` (ssh://ooicm.omaha.us.ray.com:29418/AWIPS2_baseline.git) registered for path 'submodules/AWIPS2_baseline'
- Submodule `submodules/AWIPS2_foss` (ssh://ooicm.omaha.us.ray.com:29418/AWIPS2_foss.git) registered for path 'submodules/AWIPS2_foss'
- Submodule `submodules/ogc` (ssh://ooicm.omaha.us.ray.com:29418/ogc.git) registered for path 'submodules/ogc'
- Submodule `submodules/ufcore` (ssh://ooicm.omaha.us.ray.com:29418/ufcore.git) registered for path 'submodules/ufcore'
- Submodule `submodules/ufcore-foss` (ssh://ooicm.omaha.us.ray.com:29418/ufcore-foss.git) registered for path 'submodules/ufcore-foss'

Initialized empty Git repository in `/home/thahn/git/ooi/submodules/AWIPS2_baseline/.git/`
remote: Counting objects: 272288, done
remote: Finding sources: 100% (272288/272288)
remote: Total 272288 (delta 127439), reused 269138 (delta 127439)
Receiving objects: 100% (272288/272288), 3.87 GiB | 31.45 MiB/s, done.
Resolving deltas: 100% (127439/127439), done.
Submodule path 'submodules/AWIPS2_baseline': checked out '524845d989c7568c99f6e6985a85c890baef9e17'
Initialized empty Git repository in `/home/thahn/git/ooi/submodules/AWIPS2_foss/.git/`
remote: Counting objects: 832, done
remote: Finding sources: 100% (832/832)
Receiving objects: 100% (832/832), 94.59 MiB | 26.54 MiB/s, done.
Resolving deltas: 100% (244/244), done.
remote: Total 832 (delta 244), reused 832 (delta 244)
Submodule path 'submodules/AWIPS2_foss': checked out '50e4f1b09dcd58d8eda61e6e40dd0195e82d5794'
Initialized empty Git repository in `/home/thahn/git/ooi/submodules/ogc/.git/`
remote: Counting objects: 552, done
remote: Finding sources: 100% (552/552)
Receiving objects: 100% (552/552), 541.39 KiB, done.
Resolving deltas: 100% (144/144), done.
remote: Total 552 (delta 144), reused 512 (delta 144)
Submodule path 'submodules/ogc': checked out 'bccc4af7334e44eef58d9515a3807e5c74o7e7fc26'
Initialized empty Git repository in `/home/thahn/git/ooi/submodules/ufcore/.git/`
remote: Counting objects: 6878, done
remote: Finding sources: 100% (6878/6878)
remote: Total 6878 (delta 1235), reused 6844 (delta 1235)
Receiving objects: 100% (6878/6878), 5.51 MiB, done.
Resolving deltas: 100% (1235/1235), done.
remote: Total 6878 (delta 1235), reused 6844 (delta 1235)
Submodule path 'submodules/ufcore': checked out '50b65bf9760dcde8d6a4576fca97dc31bc4c'
Initialized empty Git repository in `/home/thahn/git/ooi/submodules/ufcore-foss/.git/`
remote: Counting objects: 6316, done
remote: Finding sources: 100% (6316/6316)
remote: Total 6316 (delta 59), reused 6316 (delta 59)
Receiving objects: 100% (6316/6316), 209.89 MiB | 28.19 MiB/s, done.
Resolving deltas: 100% (59/59), done.
Submodule path 'submodules/ufcore-foss': checked out '3bc0b880e22a20002b13d2d1d98f75e7a4e16c'

**Setup Eclipse template and formatter**

1. Open Window->Preferences->Java->Code Style->Code Templates
2. Click "Import..." and browse to the ooi repo ooi/submodules/AWIPS2_baseline/edexOsgi/build.edex/opt/eclipse/codetemplates.xml then click OK
3. Then select Comments->Files from the template UI and edit to:

```java
/**
 * OOIPLACEHOLDER
 * 
 * Copyright $(year) Raytheon Co.
 **/```
4. Click Apply and OK
5. Open Window->Preferences->Java->Code Style->Formatter
6. Click "Import..." and browse to the ooi repo ooi/submodules/AWIPS2_baseline/edexOsgi/build.edex/opt/eclipse/formatter.xml then click OK
7. Click Apply and OK
8. Open Window->Preferences->General->Editors->Text Editors
9. Make sure "Insert spaces for tabs" is checked and that "Displayed tab width" is set to 4
10. Click Apply and OK

Import Projects into Eclipse

Make a new workspace

- Import all of the projects from the following directories into the workspace:

  - build
  - common
  - edex
  - features
  - foss
  - util
  - submodules/AWIPS2_baseline/edexOsgi
  - submodules/AWIPS2_baseline/ncep/gov.nasa.gsfc.fits
  - submodules/AWIPS2_foss
  - submodules/ogc
  - submodules/ufcore/common
  - submodules/ufcore/edex
  - submodules/ufcore/features
  - submodules/ufcore-foss

- Close the following projects

  - com.raytheon.edex.plugin.binligtning
  - com.raytheon.edex.rpgenvdata
  - com.raytheon.uf.common.dataplugin.ffmp
  - com.raytheon.uf.edex.plugin.ffmp

Set API baseline error reporting

1. click on 'Window->Preferences'
2. expand 'Plug-in Development'
3. click on 'API Baselines'
4. change 'Missing API baseline' option from 'Error' to 'Ignore'
5. click 'Apply'

Set JDK

1. go to Preferences ('Window->Preferences')
2. Expand 'Java'
3. click on 'Installed JREs'
4. click 'Add...'
5. select 'Standard VM' and click 'Next'
6. click 'Directory...' next to JRE home and browse to uframe java ($HOME/uframe/uframe-1.0/java)
7. name the JRE 'uframe' and click 'Finish'
8. uncheck the system java and check the uframe JRE and click 'OK'

Set up target platform location to run CAVE (one-time-only)

1. In Eclipse, select Window -> Preferences.
2. Expand Plug-in Development.
3. Select Target Platform.
4. Select Running Platform (Active).
5. Click Edit.
6. Highlight the location and click Edit again.
7. Change location to the Eclipse 3.8 instance
8. Press Finish
9. Select "Show location content" near the bottom
10. Ensure the path is valid and that plug-ins are detected.
11. Click Finish.
12. Click OK.

Deploy Edex

1. Ctrl-Shift-R, deploy-install.xml
2. Double click on the build.deploy one
3. Run As
4. Enter the following as deploy arguments
   - Change the paths as appropriate for your chosen installation location
   
   ```
   -Dgroovy.path=${env_var:HOME}/uframes/ooi/uframe-1.0/groovy/bin -Dedex.root=${env_var:HOME}/uframes/ooi/uframe-1.0/ed
   ```

Deploy extras

This deploys changes to the edex-server script and the python libraries, for the time being this is an extra step over the normal deploy above.

1. Open the deploy.xml file from raceaddons
2. Right click and "Run As -> External Tools Configuration..."
3. Click the new build icon in the top left (white paper with yellow star)
4. Enter the following as deploy arguments
   - Change the edex path as appropriate for your chosen installation location
   - Change the python path as appropriate for the location of your git clone

```
-Dedex.root.directory=${env_var:HOME}/uframes/ooi/uframe-1.0/edex
-Dpython.dir=${env_var:HOME}/workspaces/ooi_phase2/pythonPackages
```

Run Edex

1. Change to the bin directory: cd uframes/ooi/bin
2. Start the uFrame service ./edex-server all start
3. Watch the uFrame log: ./edex-server edex watch

```
***********************************************************
* EDEX ESB is now operational                             *
* Total startup time: 17.1s                               *
***********************************************************
```