

# Code Development Cycle

## Content

- [Content](#)
- [Psana-conda commands](#)
- [References](#)

## Psana-conda commands

Task	Command	Notes
Initialization of conda environment	<code>source /reg/g/psdm/bin/conda_setup</code>	Bash only for conda, no .csh
<b>Local release operations</b>		
Create local release directory	<code>condarel --newrel --name &lt;my-rel-directory&gt;</code>	In current directory creates sub-directory <my-rel-directory> with a few system files
Activate local release	<code>cd &lt;my-rel-directory&gt;</code> <code>source conda_setup</code>	
Activate release in another directory	<code>source conda_setup --reldir &lt;/path/to/another/release&gt;</code>	
Exit conda environment	<code>undo_conda</code>	
Upgrade release	<code>conda activate ana-1.3.47</code> <code>condarel --chenv</code> <code>source conda_setup</code> <code>scons -c</code> <code>scons</code>	In conda, first activate, using standard conda commands, the environment you want to build against.  Then use the --chenv command, it picks up the current conda environment.
Build code in release directory	<code>scons</code>	Being in local release directory
<b>Package operations</b>		
Checkout package from master repository	<code>condarel --addpkg --name &lt;pkg-name&gt; --tag HEAD</code>	Check out latest version of package from github (or svn) repository where package resides.
Checkout existing package from psdm users repo	<code>condarel --addpkg --user --name &lt;pkg-name&gt;</code>	condarel takes --user flag to checkout from psdm users repo.
Create new package	<code>condarel --newpkg --name &lt;new-pkg-name&gt;</code>  use github GUI to create repository for this package.	Creates the directory <new-pkg-name> with a minimal structure, include/src/app/data, and SConscript
Add pkg to psana build control	update the file <a href="#">psana-conda-svn-pkgs</a> in the directory <code>/reg/g/psdm/sw/conda/manage/config</code> that is part of the github repo <a href="#">slacslab</a> <a href="#">/anarel-manage</a>	This may be tricky due to permission to change code in <a href="#">slacslab/anarel-manage</a> .  Chris or Clemens can do that.
<b>Information commands</b>		
List available releases	<code>conda env list</code>	
Identify ana-current	<code>more /reg/g/psdm/sw/conda/current/ana/ana-current</code>	

Identify dm-current	<code>more /reg/g/psdm/sw/conda/current/dm/dm-current</code>	
Look at source code	<code>ls /reg/g/psdm/sw/conda/scratch /&lt;release&gt;/&lt;pkg&gt;</code>  <code>ls \$CONDA_PREFIX/lib/python2.7/site-packages/&lt;pkg&gt;</code>	Package source code is available through the scratch directory.  In the conda environment only python code is available.
<b>Example of regular development loop</b>		
Check in, work on package, update,  stage, commit, check out,  create new tag, etc.	<code>condarel --newrel --name &lt;my-rel-directory&gt;</code>  <code>cd &lt;my-rel-directory&gt;</code>  <code>source conda_setup</code>  <code>condarel --addpkg --name &lt;pkg-name&gt; --tag HEAD</code>  ... edit files in the package  ... run tests etc.  ... add comments about changes  <code>cd &lt;pkg-name&gt;</code>  <code>git pull --rebase</code>  <code>git status</code>  <code>git diff [module-name]</code>  <code>git add -A</code>  <code>git commit -m "comment" -a</code>  <code>git push origin master</code>  <code>git tag</code>  <code>git tag -a V01-00-26 -m "comment"</code>  <code>git push origin V01-00-26</code>	Create local release directory (once) if it does not exist  move in release directory  set and activate conda environment  add package(s) if necessary  work on code   add comments in file <pkg-name>/doc/ChangeLog  when ready to commit  update package from repo, if someone else may work on it  print info about package status  see difference in modules  stage modified modules under git control  commit changes to git  commit changes to github master repo  print available tags  create new tag  commit new tag to github repo
<b>Special cases</b>		
develop pdsdata/psalg	<code>condarel --addpkg --name pdsdata</code>  <code>condarel --addpkg --name pdsdata_ext</code>	See also <a href="#">Building the psalg and pdsdata packages</a> With conda, pdsdata and psalg are part of the psana-conda package. They get put in a subdirectory called extpkgs. You need the proxy packages to build/develop.
develop ndarray	<code>condarel --addpkg --name ndarray --tag HEAD</code> <code>condarel --addpkg --name ndarray_ext</code> scons tag ndarray when done  update version in ndarray recipe meta.yaml: <code>conda package version</code> <code>git url -- your new tag</code> build new ndarray package (admin account): <code>cd /reg/g/psdm/sw/conda/manage</code> <code>git pull</code> (or fetch? Get you edits above) <code>cd recipes/psana</code> <code>ana-rel-admin --cmd bld-pkg ndarray</code>	ndarray live in it's own conda package. The ndarray github repo does not include a SConscript. You need the ndarray_ext to link it into the build system. If ndarray_ext sees you've checked out ndarray, it overrides the ndarray in the conda environment.  After tagging your changes to ndarray, a new ndarray conda package must be built. Edit the meta.yaml on github in the recipe in the links to the right. Now from the admin account, update the management code in /reg/g/psdm/sw/conda/manage. Now build a new ndarray package with ana-rel-admin. The next time a release is built, the new ndarray is picked up.

## References

- [Psana Developer Documentation](#) - predecessor of this page created by David.
- [Common development tasks](#) - SVN based development commands, deprecated
- [Version control with git](#)