

SVN based Psana Developer Documentation

- [Introduction](#)
- [Old to New Conda Command Table](#)
- [Missing Functionality](#)
- [Converting a Release](#)

Introduction

For reference, we will keep the old SVN based page. Soon, or now, we have moved psana to github. Refer to the new page: [Psana Developer Documentation](#) for instructions using git.

This page covers documentation about the conda release and build system relevant for psana developers. This documentation is also for LCLS users that want to migrate their test releases based on psana C++ modules and the [SConsTools](#) scons build system to the conda environments.

Old to New Conda Command Table

Below we go over the steps to create, develop, build and manage test releases - comparing old commands from the RPM release system to the new conda based system.

Old commands like newrel, addpkg and relinfo are not available in the conda world. Most all steps are executed using the

condarel

program. Do condarel -h for the latest help on this script.

step	old	conda	notes
get started	source /reg/g/psdm/etc/ana_env.sh	source /reg/g/psdm/etc/ana_env.sh source conda_setup or source conda_setup	psana developers should still source the ana_env.sh so that we can easily maintain the old/RPM system as well as the new conda system. When we have deprecated the old, we can just source conda_setup. After sourcing conda_setup, /reg/g/psdm/bin is removed from your PATH it is replaced with /reg/g/psdm/sw/conda/manage/bin
create new test release directory	newrel ana-current myrel	condarel --newrel --name myrel	in conda, myrel is based on the current conda environment. The previous conda_setup command activated a conda environment like ana-1.0.7. If you are not in a conda environment with psana-conda installed, condarel will fail. (sourcing conda_setup will automatically activate such an environment) The old myrel directory has the hidden file .sit_release with content like ana-0.19.21 In the new myrelease directory, the content will be the psana-conda package name and version. The new myrel directory will also have the hidden file .sit_conda_env with the full path of the conda environment myrelease is built against
activate test release	cd myrel sit_setup	cd myrel source conda_setup	conda_setup looks in the current directory (like sit_setup) for the special files mentioned above (but see row below). It sets PATH, LD_LIBRARY_PATH and PYTHONPATH to first look for programs, libraries and python modules built in your test release before looking for them in the conda environment. conda_setup will add *tr* to your prompt to indicate that you are in a test release .
activate test release in another directory	sit_setup /path/to/my/release	source conda_setup --reldir /path/to/my/release	As above, but activate a test release in another directory
create new package	newpkg MyPkg	condarel --newpkg --name MyPkg	Creates the directory MyPkg with a minimal structure, include/src/app/data, and SConsript

create new package in psdm svn repo	psvn newpkg MyPkg	same	nothing created in your test release, just starting a package in the svn psdm repo
create new package in psdm users repo	psvn -u newpkg MyPkg	same	psvn still works the same, as above, nothing created in your test release, this starts a package in the svn users repo
checkout new package	addpkg MyPkg	condarel --addpkg --name MyPkg	for packages that are not part of psana-conda gets it from head
checkout existing package	addpkg Xtclnput	condarel --addpkg --name Xtclnput	Looks up the appropriate tag for the version of psana-conda. Checks out that tag
checkout existing package from psdm users repo	addpkg -u MyPkg	condarel --addpkg --user --name MyPkg	condarel takes --user flag to checkout from psdm users repo. You can also use --tag if you maintain tags in your repo.
checkout from HEAD	addpkg Xtclnput HEAD	condarel --addpkg --name Xtclnput --tag HEAD	You can also specify specific tags with the --tag argument.
build	scons	scons	same
develop pdsdata/psalg or ndarray	very awkward	condarel --addpkg --name pdsdata condarel --addpkg --name pdsdata_ext	With conda, pdsdata, psalg and ndarray are part of the psana-conda package. They get put in a subdirectory called extpkgs. You then also need the proxy packages to build. Note - the pdsdata_ext SConscript make command does not generate the dll targets - you may need to set environment variables and run make yourself for certain development.
release info	relinfo	condarel --relinfo	
upgrade release	relupgrade ana-19.0.20 sit_setup scons -c scons	source activate ana-1.0.8 condarel --chenv source conda_setup scons -c scons	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.
develop Translator	addpkg Translator scons	condarel --addpkg --name Translator condarel --addpkg --name hdf5 condarel --addpkg --name openmpi scons	Since the Translator includes headers using package names, i.e., #include "hdf5/hdf5.h" You must first include the hdf5 and openmpi proxy packages
work with SConsTools		condarel --addpkg --name SConsTools	In conda we get SConsTools from the conda branch. Don't specify --name HEAD or a tag, let condarel get the latest version of the code from the conda branch.
add it	addpkg SConsTools		
remove it	rm -r SConsTools	rm -r SConsTools rm SConstruct condarel --sconslnk	If you remove SConsTools after checking it out, you have to do additional steps to restore SConstruct. Namely, removing the SConstruct link to the previously checked out version, and then using the --sconslnk command in condarel. This creates the link SConstruct --> SConstruct.main installed in conda env.
Test	scons test	scons test	
work on package check in new tag to svn psdm repo	addpkg MyPkg HEAD cd MyPkg # modify code svn status # see summary svn diff # see changes svn update svn commit -m "message" psvn tags psvn tag V00-00-00	condarel --addpkg --name MyPkg --tag HEAD same. same same same same same same same	the psvn program functions as before, you can run psvn tags to see all the tags, then psvn tag to create a new tag.
track diffs	svn diff -r7810:HEAD file.h	same	There is no change to how you use svn commands like svn diff, this is an example of diffing a file, file.h, that is checked into head, against a previous revision
exit conda	--	undo_conda	If you need to get out of the conda world, and go back to where you were before (rpm based psana, if you are sourcing /reg/g/psdm/etc/ana_env.sh) then the undo_conda command does this. NOTE: releases built in the conda world will not work in the old RPM world.
list releases	ls \$SIT_RELDIR	conda env list	Think of the old RPM based releases as conda environments - use standard conda commands to see them note - this lists your own environments (if you've made any) in addition to the ana environments maintained at LCLS.

identify ana-current	ls -l \$SIT_RELDIR/ana-current	cd /reg/g/psdm/sw/conda/ana-current cat ana-current	NOTE: this is a implementation detail, how we identify which conda environment is 'ana-current' is subject to change
add pkg to psana	edit the file ana-tags in the /reg/g/psdm/sw/releases/buildbot /tags directory	update the file psana-conda-svn-pkgs in the directory /reg/g/psdm/sw/conda/manage/config that is part of the github repo anarel-manage	We should document this more completely in the Admin Documentation

Missing Functionality

Not all functionality of addpkg, relinfo, sit_setup etc have been implemented. If there is a feature that you need, let me know.

If you want to use the old commands, i.e, addpkg instead of condrel --addpkg, we can write new wrappers - but I think while we transition it is good to keep the interfaces distinct as psana developers will be working with both build systems.

Converting a Release

I recommend leaving an old release alone and starting new ones based on conda, however there are two commands in condarel to convert back and forth. This should work for users writing their own C++ psana modules, but won't work for psana developers that have checked out certain external proxy packages. These commands are

```
condarel --convert2conda
```

note the name of the old rpm release that you lost. If you want to switch back, use that name with the

```
condarel --convert2rpm
```

command. See condarel -h for details.