

psana - Quick Start



Unknown macro: 'html'

Recommended Next Topics

- [psana - User Manual](#)

Other Related Topics

- [Account Setup](#)



Unknown macro: 'html'



Unknown macro: 'html'

- [About](#)
- [1. Create a test release](#)
- [2. Change to the test release](#)
- [3. Create new analysis package \(Freq: once\)](#)
- [4. Add analysis module \(Freq: once\)](#)
- [5. Modify analysis code \(Freq: as much as you need\)](#)
- [6. Build \(Freq: after every edit\)](#)
- [7. Create configuration file \(Freq: once\)](#)
- [8. Run the analysis \(Freq: as much as you need\)](#)



Unknown macro: 'html'

About

This document explains which steps need to be done to begin using [psana modules](#). Once you have completed the following you can view a list of more advanced command in the section [Common development tasks](#).

1. Create a test release

At this step you will be using the release creation tool to create and properly initialize your test release directory. The directory should be created at a file system to which you have the write access. Releases are explained in [Packages and Releases](#) section.

The `newrel` command requires two positional parameters: the name of a base release (in this example: **ana-current**) and the name of a directory which will be created for your test release (in this example: **analysis-rel**):

```
newrel ana-current analysis-rel
```

This operation will create the directory and populate it with two files:

```
% ls analysis-rel/  
SConstruct  
.sit_release
```

Other directories and hidden system files may appear later at compilation stage or when you run `addpkg` or `newpkg` commands.

Below we refer to this directory as a "test release". Usually separate analyses use separate test releases, and each test release should bear unique name if they are located in the same directory.

2. Change to the test release

Now you need to change your working directory to the newly created one where your test release is located. Then you will run **sit_setup** to make proper adjustments to your environment variables so that will recognize the fact that you'll be working with the test release. Note that some of those variables will be relative to the test release directory, so that you may be required to execute certain operations while being within that directory. If you choose to leave the directory then you may want to run `sit_setup` again to set up a different execution environment.

These two command should be done just once when you're about to work with that test release.

```
cd analysis-rel
sit_setup
```

3. Create new analysis package (Freq: once)

Choose sufficiently unique name for your analysis package and create the package.



It's better to choose name which contains letters, digits, and underscores only because the package name will be used for Python package names.

```
newpkg my_ana_pkg
```

This will create directory `my_ana_pkg` and populate it with few files and sub-directories:

```
.../analysis-rel/
    ./my_ana_pkg/
        SConscript
        doc/
            README
            ChangeLog
```

4. Add analysis module (Freq: once)

Execute following commands:

```
mkdir my_ana_pkg/src my_ana_pkg/include # only if src and include directories do not exist yet
codegen -l psana-module my_ana_pkg my_ana_mod
```

For a python module:

```
mkdir my_ana_pkg/src # only if src directory does not exist yet
codegen -l python my_ana_pkg my_ana_mod
```

The above will create "skeleton" files for an analysis module.



Like with the package name it's better to use names containing letters, digits, and underscores only.

5. Modify analysis code (Freq: as much as you need)

Use your favorite editor to add or modify code the analysis module. For C++ modules see [Psana User Manual](#) for examples.

6. Build (Freq: after every edit)

Run the command which builds all necessary software in your test release:

```
scons
```

7. Create configuration file (Freq: once)

In addition to analysis module the job usually needs a separate configuration file. The name and location of the file could be arbitrary. C++-based analysis uses file named `psana.cfg` in current directory by default, create it with your favorite editor:

```
vi psana.cfg
```

The file will contain both the names of the modules you want to load and parameters you want to pass to those modules, exact content of the configuration file is explained in [Psana User Manual](#).

8. Run the analysis (Freq: as much as you need)

Chose the data you want to process and start analysis job. Assuming properly prepared psana.cfg in current directory, execute:

```
psana /reg/d/psdm/AMO/amo14410/xtc/e23-r0406-*.xtc
```

or:

```
psana exp=amo14410:run=0406
```