

NAME

snoplus-env – script for setting up SNO+ software environments

SYNOPSIS

source snoplus-env [**-h**misl] [**-d** *path*] [**-f** *exec-file*] [*rat-version* | *label*]

Deprecated, kept for legacy compatability:

source snoplus-env help
source snoplus-env man[ual]
source snoplus-env info | **env**[ironment]
source snoplus-env snoing
source snoplus-env list
source snoplus-env *path*

DESCRIPTION

snoplus-env provides a simple interface for correctly setting up the environment for the requested version. This script is intended to be sourced.

OPTIONS

If no options are given, **SNOPLUS_DEFAULT_ENV** is used. Listed below includes both the preferred and deprecated, legacy options.

rat-version

Any version or RAT that is installed via **snoing.py**.

label A label corresponding to a certain version of RAT. Characters within parentheses are optional, within brackets dictate a choice.

– Production version labels:

perylene(prod(uction))(numu)(1)
 Perylene Production $\nu\mu$, RAT 5.2.2

solar(prod(uction))(1)
 Solar Production, RAT 5.2.3

partial water(fill)(prod(uction))(1)
 Partial Water Fill Production, RAT 5.3.1

water(fill)(prod(uction)) 1
 Water Fill Production, RAT 5.3.2

water(fill)(prod(uction))(2)
[Tt]e-[Dd]iol(prod(uction)) 1
water(fill)(and) [Tt]e-[Dd]iol(prod(uction))(1)
 Water and Te-Diol Production, RAT 6.1.1

[Tt]e-[Dd]iol(prod(uction))(2)
 Te-Diol Production, RAT 6.1.2

proc(essing) challenge 1
 Processing Challenge, RAT 6.2.0

proc(essing) challenge 2
 Processing Challenge (bugfix), RAT 6.2.1

proc(essing) challenge 3
 Processing Challenge update, RAT 6.2.2

proc(essing) challenge(4)
 Processing Challenge update, RAT 6.2.3

– Non-production version labels:

latest | new(est)

The latest RAT release version installed via **snoing.py**.

local(dev(elop(ment)))

Local development, this is the default if no argument is given, falling back on the global development version if no local exists, and **RATDEV_FALLBACK** is set (default is set).

global(dev(elop(ment)))

Global development, this is in-line with the current development version of RAT (perhaps a day behind). This should not be considered stable.

-d path

path A path to a version of RAT; treated as if calling for a local development copy, with **RATDEV** set as the given path.

-h

help Print basic help.

-m**man[ual]**

Open this manual by loading the environment first, to ensure it is in your **MANPATH**.

-i**info | env[ironment]**

Prints relevant environment variables.

-s

snoing Load the snoing environment, rather than the RAT environment; intended for installing software by the SNO+ admin. Regular users should not need to use this.

-l

list List all the versions of RAT installed via **snoing.py**.

-f exec-path

An executable, like a bash script, to be executed from within the SNO+ environment, and exit on completion.

EXIT STATUS

0	Exited without errors.
1	Exited with unspecified error.
10	Invalid option.
11	Missing argument to option.
20	Some required environment variable has not been defined.
21	Something in the environment cannot be accessed, either directory does not exist, or user has incorrect permissions.
22	Attempted operation is in wrong environment.
30	SNOING is undefined.
31	SNOING cannot be accessed, either directory does not exist, or user has incorrect permissions.
32	SNOING does not contain a snoing installation.
41	Requested version of RAT cannot be accessed, either directory does not exist, or user has incorrect permissions.
42	Requested version of RAT is not installed.

ENVIRONMENT

The following variables are intended for the user/admin to set to override the defaults.

RATDEV

If you are using a local development version, this should point to its path.

RATDEV_FALLBACK

If set, and **RATDEV** is not defined, and a local development version is requested, the environment falls-back to using the global development version of RAT.

It is recommended to be set by default in `$$SNOPLUS_ROOT/etc/snoplus` allowing the user can override this by setting **RATDEV_FALLBACK** to NULL; for example,

```
$ export RATDEV_FALLBACK=
```

SNOING

Refers to the location of snoing location. Defaults to `$$SNOPLUS_ROOT/opt/snoing`.

SNOPLUS_DEFAULT_ENV

Determines the default environment to load when no arguments given to **snoplus-env** (default is local development version).

SNOPLUS_GROUP

On most systems all SNO+ users will have a common group. If set, it is used to ensure in the environment, the user's active group is **SNOPLUS_GROUP**.

SNOPLUS_PYTHON_VENV_SCRIPT

If you are using a Python Virtual Environment (via **virtualenv**), this should point to the *activate* script.

SNOPLUS_ROOT

This refers to the location of the SNO+ software root directory **snoing.py** allows for multiple versions of the SNO+ software to be installed in parallel.

If this is not set, default is chosen by locating the closest direct ancestor directory containing an *etc/* subdirectory.

It is important that you do not set the following variables:

```
SNOPLUS_ENV      SNOPLUS_ENV_SCRIPT    SNOPLUS_ENV_ARGS    SNO-  
PLUS_RUN_FILE
```

FILES

`$$SNOPLUS_ROOT/etc/snoplus`

Configuration file to store system default environment variables. This file is sourced fairly early in the script.

`$$SNOPLUS_ROOT/etc/snoplus.d/*.sh`

It is suggested that the `$$SNOPLUS_ROOT/etc/snoplus.d` directory is created for other SNO+ environment scripts to be sourced and/or executed when **snoplus-env** is sourced. It is suggested that the following be appended to `$$SNOPLUS_ROOT/etc/snoplus`.

```
if [[ -d $$SNOPLUS_ROOT/etc/snoplud.d && -r $$SNOPLUS_ROOT/etc/snoplus.d ]];  
  for script in $$SNOPLUS_ROOT/etc/snoplud.d/*.sh; do  
    [[ -r $script ]] && source $script  
  done  
fi
```

Alternatively, you can use the utilities provided in `$$SNOPLUS_ROOT/init/snoplus::utils`:

```
source snoplus::utils  
snoplus::utils::source_dir $$SNOPLUS_ROOT/etc/snoplus.d "*.sh"
```

EXAMPLE

In the following examples, you will be presented with an example, the code to execute, and its output (at time of writing this section); the environment variables in some of the outputs will be expanded.

Load your local development version, default action:

```
$ source snoplus-env
Loaded RAT local_dev:$RATDEV
```

Load RAT 5.2.2 explicitly:

```
$ source snoplus-env 5.2.2
Loaded RAT 5.2.2
```

Load the latest Water Fill Production version of RAT, using a label:

```
$ source snoplus-env water
Loaded RAT 5.3.2
```

Equivalently, the option given could have been:

5.3.2, water, water prod, water fill production 1, etc.

Load a local development RAT using a given path:

```
$ source snoplus-env -d ~/development/rat
Loaded RAT local_dev:$HOME/development/rat
```

Load the latest version of RAT installed via **snoing.py**:

```
$ source snoplus-env latest
Loaded RAT 6.1.0
```

List all versions of RAT installed via **snoing.py**:

```
$ source snoplus-env -l
Installed versions of RAT:
- 5.2.2
- 5.2.3
- 5.3.1
- 5.3.2
- 6.1.0
- dev
- local_dev:$RATDEV
```

Run a script of RAT macros over the latest Solar Production version of RAT:

```
$ source snoplus-env -f ~/development/suite_of_macros.sh solar
Loaded RAT 5.2.3
... relevant script output
```

NOTES

Further SNO+ Lancaster documentation can be found on the SNO+ Lancaster website. (<http://py-box.lancs.ac.uk/neutrino/snoplus>)

The latest source and documentation can be found on the SNO+ Lancaster Repository. (<https://github.com/parnmatt/snoplus-lancs>)

BUGS

- This script is only compatible with **bash** and **zsh** shells.
- There is an issue with executing `source snoplus-env -h` multiple times from within the same **zsh** session.

If a bug is found, please contact the author directly, or file an issue or pull request to the SNO+ Lancaster Repository. (<https://github.com/parnmatt/snoplus-lancs>)

AUTHOR

Matt Parnell <m.parnell@lancaster.ac.uk>