
Nagios Plug-in Process Controller Documentation

Release 0.1.1

Maarten

Mar 10, 2018

Contents:

1	Nagios Plug-in Process Controller	1
1.1	Features	1
1.2	Credits	2
1.3	References	2
2	Installation	3
2.1	Stable release	3
2.2	From sources	3
3	Usage	5
4	Contributing	7
4.1	Types of Contributions	7
4.2	Get Started!	8
4.3	Pull Request Guidelines	9
4.4	Tips	9
4.5	Deploying	9
5	Credits	11
5.1	Development Lead	11
5.2	Contributors	11
6	History	13
6.1	0.1.0 (2018-03-10)	13
7	Indices and tables	15

Nagios Plug-in Process Controller

NPPC, Nagios Plug-in Process Controller, is a set of scripts and configuration files that let you periodically runs Nagios Plug-in parallel using `systemd` and `systemd.timer`. Results can be posted via HTTPS to a NSCAweb server.

NPPC consists of:

- A wrapper around Nagios Plug-in that controls the output, exit status and time-outs to have it safely run by a `systemd.timer`
- A script, `systemd.service` script and configuration file that creates the `systemd.timer` files.
- A script, `systemd.timer` and configuration file that sends periodically output to NSCAweb
- Free software: <https://opensource.org/licenses/ISC>
- Documentation: <https://nppc.readthedocs.io>.
- GitHub: <https://github.com/maartenq/nppc>
- PyPi: <https://pypi.python.org/pypi/nppc>
- Travis CI: <https://travis-ci.org/maartenq/nppc>
- Codecov: <https://codecov.io/github/maartenq/nppc>

1.1 Features

- Parallel execution of Nagios Plug-ins.
- Termination of Nagios Plug-ins if maximum time exceeds.
- Posts check results external commands.
- Command definition in YAML or text format.
- Configuration in YAML.
- Simple modular implementation.

1.2 Credits

This package was created with [Cookiecutter](#) and the [audreyr/cookiecutter-pypackage](#) project template.

1.3 References

2.1 Stable release

To install Nagios Plug-in Process Controller, run this command in your terminal:

```
$ pip install nppc
```

This is the preferred method to install Nagios Plug-in Process Controller, as it will always install the most recent stable release.

If you don't have `pip` installed, this [Python installation guide](#) can guide you through the process.

2.2 From sources

The sources for Nagios Plug-in Process Controller can be downloaded from the [Github repo](#).

You can either clone the public repository:

```
$ git clone git://github.com/maartenq/nppc
```

Or download the [tarball](#):

```
$ curl -OL https://github.com/maartenq/nppc/tarball/master
```

Once you have a copy of the source, you can install it with:

```
$ python setup.py install
```


CHAPTER 3

Usage

To use Nagios Plug-in Process Controller in a project:

```
import nppc
```


Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given. You can contribute in many ways:

4.1 Types of Contributions

4.1.1 Report Bugs

Report bugs at <https://github.com/maartenq/nppc/issues>.

If you are reporting a bug, please include:

- Your operating system name and version.
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

4.1.2 Fix Bugs

Look through the GitHub issues for bugs. Anything tagged with “bug” and “help wanted” is open to whoever wants to implement it.

4.1.3 Implement Features

Look through the GitHub issues for features. Anything tagged with “enhancement” and “help wanted” is open to whoever wants to implement it.

4.1.4 Write Documentation

Nagios Plug-in Process Controller could always use more documentation, whether as part of the official Nagios Plug-in Process Controller docs, in docstrings, or even on the web in blog posts, articles, and such.

4.1.5 Submit Feedback

The best way to send feedback is to file an issue at <https://github.com/maartenq/nppc/issues>.

If you are proposing a feature:

- Explain in detail how it would work.
- Keep the scope as narrow as possible, to make it easier to implement.
- Remember that this is a volunteer-driven project, and that contributions are welcome :)

4.2 Get Started!

Ready to contribute? Here's how to set up *nppc* for local development.

1. Fork the *nppc* repo on GitHub.
2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/nppc.git
```

3. Install your local copy into a virtualenv. Assuming you have virtualenvwrapper installed, this is how you set up your fork for local development:

```
$ mkvirtualenv nppc
$ cd nppc/
$ python setup.py develop
```

4. Create a branch for local development:

```
$ git checkout -b name-of-your-bugfix-or-feature
```

Now you can make your changes locally.

5. When you're done making changes, check that your changes pass flake8 and the tests, including testing other Python versions with tox:

```
$ flake8 nppc tests
$ python setup.py test or py.test
$ tox
```

To get flake8 and tox, just pip install them into your virtualenv.

6. Commit your changes and push your branch to GitHub:

```
$ git add .
$ git commit -m "Your detailed description of your changes."
$ git push origin name-of-your-bugfix-or-feature
```

7. Submit a pull request through the GitHub website.

4.3 Pull Request Guidelines

Before you submit a pull request, check that it meets these guidelines:

1. The pull request should include tests.
2. If the pull request adds functionality, the docs should be updated. Put your new functionality into a function with a docstring, and add the feature to the list in README.rst.
3. The pull request should work for Python 2.7, 3.4, 3.5 and 3.6, and for PyPy. Check https://travis-ci.org/maartenq/nppc/pull_requests and make sure that the tests pass for all supported Python versions.

4.4 Tips

To run a subset of tests:

```
$ py.test tests.test_nppc
```

4.5 Deploying

A reminder for the maintainers on how to deploy. Make sure all your changes are committed (including an entry in HISTORY.rst). Then run:

```
$ bumpversion patch # possible: major / minor / patch
$ git push
$ git push --tags
```

Travis will then deploy to PyPI if tests pass.

5.1 Development Lead

- Maarten <ikmaarten@gmail.com>

5.2 Contributors

None yet. Why not be the first?

6.1 0.1.0 (2018-03-10)

- Project setup with `'audreyr/cookiecutter-pypackage'` project template.

CHAPTER 7

Indices and tables

- `genindex`
- `modindex`
- `search`