

Python-Module per RPM verteilen

Categories : [Linux](#), [OSBN/Ubuntuusers Planet](#), [XING / LinkedIn](#)

Date : 30. September 2016

Im [Python Package Index \(PyPI\)](#) können zahlreiche Python-Module in Form von Quellcode bezogen werden. Die Installation dieser Module erfolgt in der Regel über die Ausführung eines Installationskripts, nachdem das entsprechende Archiv heruntergeladen und entpackt wurde. Mithilfe des Paket-Managers **pip** (*pip installs packages*) kann dieser Vorgang automatisiert werden - beispielsweise für das Modul [nagiosplugin](#):

```
# pip install nagiosplugin
```

Wenn man größere Systemlandschaften pflegt ist das in der Regel keine denkbare Option, da sich so installierte Module nicht über ein zentrales Software-/Patch-Management überwachen und steuern lassen. Eine schönere Option wäre es, die Module per **RPM** zu verteilen. Glücklicherweise liefern die Installationskripte der meisten Python-Module hierfür schon [vorgefertigte Routinen](#) mit.

Das folgende Beispiel erstellt für das oben genannte Modul eine RPM-Datei:

```
$ wget https://pypi.python.org/packages/source/n/nagiosplugin/nagiosplugin-1.2.4.tar.gz
$ tar xf nagiosplugin-1.2.4.tar.gz ; cd nagiosplugin-1.2.4
$ python setup.py bdist_rpm
running bdist_rpm
running egg_info
... moving build/bdist.linux-x86_64/rpm/SRPMS/nagiosplugin-1.2.4-1.src.rpm -> dist
moving build/bdist.linux.x86_64/rpm/RPMS/noarch/nagiosplugin-1.2.4-1.noarch.rpm -> dist
```

Das **python**-Kommando muss im Ordner, in welchem der Quellcode entpackt wurde, ausgeführt werden. Schlägt dieser Befehl fehl (*Option/Parameter ungültig*), sollte der folgende Befehl zum Ziel führen:

```
$ python setup.py bdist --formats=rpm
```

Anschließend befinden sich im Ordner **dist** die entsprechenden RPM-Dateien (*Quellcode-Paket, installierbares Paket*), sowie der Quellcode des Moduls. Im Ordner **build** befinden sich Dateien, die während der Übersetzung entstehen, sowie die RPM-Spezifikation. Diese herkömmliche Spezifikation kann dann beispielsweise dafür verwendet werden, das Paket über **Continuous Integration** für mehrere Architekturen und Distributionen zu erstellen. :)

Wer nur die RPM-Spezifikation erstellen möchte, ohne gleich ein RPM-Paket zu erstellen, ändert den oben genannten Befehl wie folgt ab:

```
$ python setup.py bdist_rpm --spec-only
```

Bei manchen Linux-Distributionen erscheint beim Übersetzen gelegentlich der folgende Fehler:

```
UnicodeDecodeError: 'ascii' codec can't decode byte ... in position ...: ordinal not in range(128)
```

Der Fehler kommt meistens dadurch zustande, dass in der Dokumentation des Moduls **non-Unicode** Zeichen verwendet werden. Während der Erstellung der RPM-Spezifikation bricht das Python-Installationsskript mit einem Fehler ab (*sollte das nicht eigentlich abgefangen werden können?!*). Abhilfe schafft es, mit **grep** auf die Suche nach den entsprechenden Zeichen zu gehen und diese anzupassen bzw. zu entfernen:

```
$ grep -P "[\x80-\xFF]" *.txt CONTRIBUTORS.txt:* Simónè Gíèrtz
```