

Software Collections

When you need *all* the versions

Jan Staněk <jstanek@redhat.com> – Software Collections SIG
CentOS Dojo in Brussels 2020

- Associate Software Engineer @ Red Hat
- Co-maintainer of NodeJS collections from Red Hat
- Maintainer of **softwarecollections.org**
- Contact person for the Software Collections (SCLo) SIG



What is a Software Collection

Goals

Software Collections give you the power to build and concurrently install multiple versions of the same components on your system, without affecting the system versions of the packages installed from your distribution.

- Distribute multiple versions of software as RPMs.
- Allow projects to move at a different pace than the whole distribution.
- Give users more flexibility and choice in software selection.

So is it the same thing as AppStream?

- SCLs pre-date Modularity/AppStream concepts.
 - SCLs can be (in a way) considered a prototype for AppStream.
- SCLs are currently available for CentOS 6 and 7.
 - AppStream is CentOS 8/Stream only.
- The creation and build processes differ significantly.

SCL or AppStream

SCL

CentOS 6, 7, (8)

Parallel Installability

Different FS paths

Special RPM macro magic

AppStream

CentOS 8/Stream only

Parallel Availability

Regular FS paths

Special build system

Where to get one

- Repositories available in `centos-release-scl` package for CentOS 6 and 7.
- CentOS 7 language containers (`centos/python-36-centos7`) provide the relevant SCL already installed and enabled.



Using a Software Collection

Typical usage

- Install the collection (or specific packages):

```
# yum -y install rh-python36
```

- Run commands in collection environment:

```
$ scl enable rh-python36 -- python --version
```

```
$ scl enable rh-python36 -- $SHELL
```

```
[root@85ac05a9da9a ~]# yum -q -y install centos-release-scl
warning: /var/cache/yum/x86_64/7/extras/packages/centos-release-scl-2-3.el7.cent
Public key for centos-release-scl-2-3.el7.centos.noarch.rpm is not installed
Importing GPG key 0xF4A80EB5:
  Userid      : "CentOS-7 Key (CentOS 7 Official Signing Key) <security@centos.org
  Fingerprint: 6341 ab27 53d7 8a78 a7c2 7bb1 24c6 a8a7 f4a8 0eb5
  Package    : centos-release-7-7.1908.0.el7.centos.x86_64 (@CentOS)
  From       : /etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-7
[root@85ac05a9da9a ~]# yum -q -y install rh-python36 rh-postgresql12
warning: /var/cache/yum/x86_64/7/centos-sclo-rh/packages/rh-postgresql12-3.4-1.e
Public key for rh-postgresql12-3.4-1.el7.x86_64.rpm is not installed
Importing GPG key 0xF2EE9D55:
  Userid      : "CentOS SoftwareCollections SIG (https://wiki.centos.org/SpecialIn
  Fingerprint: c4db d535 b1fb ba14 f8ba 64a8 4eb8 4e71 f2ee 9d55
  Package    : centos-release-scl-rh-2-3.el7.centos.noarch (@extras)
  From       : /etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-SIG-SCLO
[root@85ac05a9da9a ~]# scl enable rh-python36 -- python --version
Python 3.6.9
[root@85ac05a9da9a ~]# scl enable rh-python36 rh-postgresql12 -- $SHELL
[root@85ac05a9da9a ~]# python --version
Python 3.6.9
[root@85ac05a9da9a ~]# psql --version
psql (PostgreSQL) 12.1
[root@85ac05a9da9a ~]# exit
[root@85ac05a9da9a ~]# █
```

Structure of a collection package

Collection packages usually have common name prefix, and install into separate root at `/opt`.

Example: `rh-python36-python`

- has the common prefix `rh-python36`¹
- installs files into `/opt/rh/rh-python36/root`
 - `/opt/rh/rh-python36/root/usr/bin/python`
 - `/opt/rh/rh-python36/root/usr/bin/python3`
 - `/opt/rh/rh-python36/root/...`

¹Usually the name of the collection.

Metapackage

A *metapackage* ships the necessary plumbing for collection usage, and usually installs other collection packages as dependencies.

The plumbing is mostly a single `enable` shell script that modifies `$*PATHs` (`$PATH`, `$LDPATH`, ...).

Example: `rh-python36`

- requires `rh-python36-python` (and others)
- ships `/opt/rh/rh-python36/enable`:

```
export PATH=/opt/rh/rh-python36/root/usr/bin${PATH:+:${PATH}}
export MANPATH=/opt/rh/rh-python36/root/usr/share/man:${MANPATH}
...
```



The Life of a Collection

<https://www.softwarecollections.org/en/docs/guide/>

1. Decide on name and root:

```
name {prefix}-{app}{version}
root /opt/{provider}/{name}/
```

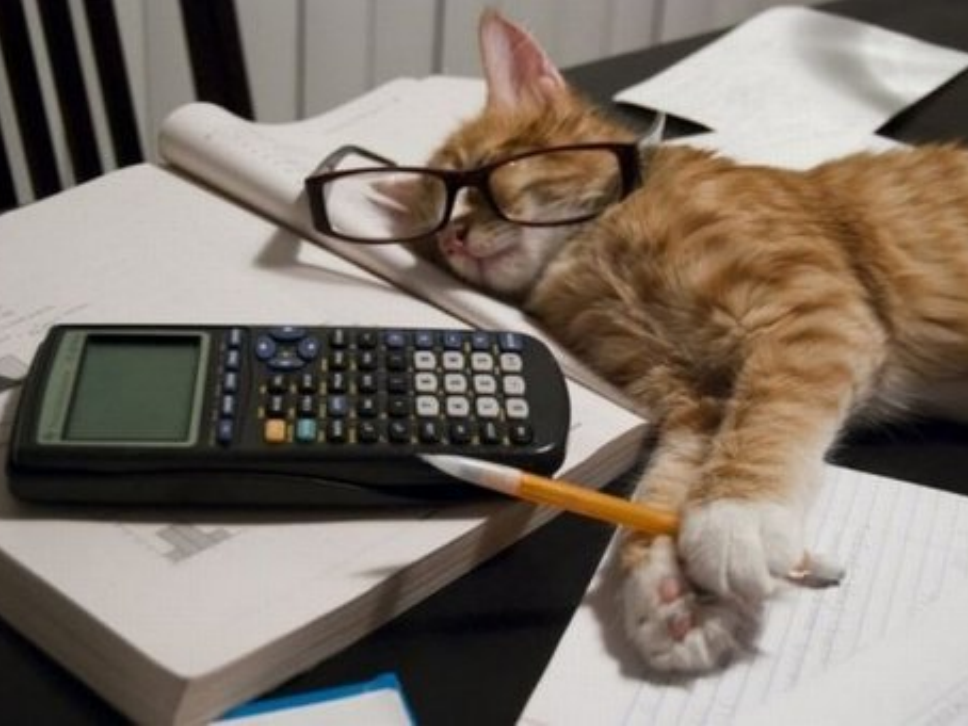
2. Create metapackage.
3. Create or convert the rest of the packages.
 - `$ pip install --user spec2scl # TODO: EPEL7`
 - `%scl_package,%{?scl}`, and friends

<https://wiki.centos.org/SpecialInterestGroup/SCLo>

1. Become SCLo SIG member.
 - `centos-devel@centos.org`
2. Request build target and tags in CBS.
 - Build target for each major CentOS release
 - 3 tags per target ending in `candidate/testing/release`
3. Build the collection in CBS.
 - `$ pip install --user rpm-list-builder`
4. Write some tests and enable CI.
 - `github.com/sclorg/sclo-ci-tests`

<https://wiki.centos.org/SpecialInterestGroup/SCLO>

1. Tag at least 1 package in each synchronized tag.
 - Usually the metapackage.
2. Request synchronization to buildlogs and mirrors.
 - `git.centos.org/centos/cbs-content-control`
3. Announce and celebrate collection availability!
 - Announce to `centos-announce@centos.org`.
 - List your collection at `softwarecollections.org`.



SIG status and future

Collections in CentOS 8/Stream

- Most RH collections replaced by AppStreams
- Developer toolsets will remain collections
- Rest is up to the community

How to get involved

- *Use*: Ask on ML, come to IRC meetings, report bugs.
 - `sclorg@redhat.com` mailing list
 - `#centos-meeting` at Freenode every 2 weeks on Tuesday, 15:00 Europe/Prague.
 - `bugs.centos.org`, SCLo project
- *Create*: As above, new SIG members welcome!
- *Automate*: Share working script, propose test enhancements, ...



Q/A

Thank you for coming!