

Wichtige Repositories unter CentOS 7

Von [Fedora](#) wird das Projekts [Extra Packages for Enterprise Linux](#) kurz EPEL betrieben. Dieses Repository enthält nur Programme (aktuell über 5.600 RPM Pakete), die im RHEL und Centos Grundpaket nicht enthalten sind. In der Regel sind die Pakete auch aktueller im Vergleich zum RPMForge-Repository.



RPM PACKAGE
MANAGEMENT

EPEL Repository einbinden

EPEL (Extra Packages for Enterprise Linux) is open source and free community based repository project from Fedora team which provides 100% high quality add-on software packages for Linux distribution including RHEL (Red Hat Enterprise Linux), CentOS, and Scientific Linux. EpeL project is not a part of RHEL/Cent OS but it is designed for major Linux distributions by providing lots of open source packages like networking, sys admin, programming, monitoring and so on. Most of the epel packages are maintained by Fedora repo.

Intallation:

```
# yum -y install  
http://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm
```

SCL (Software Collection) Repository einbinden

Bei den SCL Versionen (*ursprünglich von Redhat*), werden alle installierten Pakete in einem eigens nachgebauten root-directory unter `/opt/rh` installiert und konfiguriert.

Applikation, aus der SCL auch von der Prompt aus zugreifbar machen..

Soll ein bestimmtes SCL-Paket z.B. nun das php70 Paket auch von der Bash aus erreichbar sein, so müssen die Variablen dieses Paketes mit denen des Systems abgeglichen werden.

Dazu wird folgendermassen vorgegangen:

1. Installation der gewünschten Collection:

```
# yum install rh-php70
```

2. Registrieren der Systemvariabeln der Collection:

```
# scl enable rh-php70 bash
```

RPMForge Repository einbinden

RPMforge repository is a utility that is used to install third party software packages under **Red Hat Enterprise Linux** (RHEL) and **Community ENTERprise Operating System** (CentOS). It provides more than 5000 software packages in the rpm format for these Linux distributions.

RPMforge repository not a part of RHEL or CentOS but it is designed to work with these operating systems. The complete list of RPMForge packages can be found at <http://packages.sw.be/>.

Befehl:

```
# rpm -Uvh  
http://pkgs.repoforge.org/rpmforge-release/rpmforge-release-0.5.3-1.el7.rf.x  
86_64.rpm
```

REMI Repository einbinden

The **Remi** repository provides newer versions of the software in the core CentOS and Red Hat Enterprise Linux repositories. The Remi repository depends on the EPEL repository.

ACHTUNG: *Package names in the Remi repository are the same as the package names used in the official repositories. This similarity can result in **inadvertent package upgrades** when you run an update with yum, so use the Remi repository with care.*

Befehl:

```
# rpm -Uvh http://rpms.famillecollet.com/enterprise/remi-release-7.rpm
```

Installation von Priorities (optional)

Dies kann sowohl für die automatische als auch die manuelle Installation durchgeführt

werden.

Das Paket **priorities** wird gebraucht, um zu entscheiden; welche Pakete die neusten und stabilsten auf den verschiedenen Repositories sind. Diese bekommen dann eine höhere Priorität und es werden dann auch nur noch jene auf den Systemen installiert, welche für den produktiven Einsatz vorgesehen sind. Installiert wird das Paket folgendermassen.

```
# yum install yum-plugin-priorities -y
```

```
# vim /etc/yum/pluginconf.d/priorities.conf
```

Anschliessend prüfen wir, ob in der Konfigurationsdatei **/etc/yum/pluginconf.d/priorities.conf** folgende Zeilen eingetragen sind:

```
[main]
enabled = 1
```

Als nächstes werden die Prioritäten der Repositories in der Konfig-Datei **/etc/yum.repos.d/CentOS-Base.repo** definiert:

```
# vim /etc/yum.repos.d/CentOS-Base.repo
```

Von Bedeutung ist hierbei jeweils die Zeile:

```
priority=n
```

n ist eine Zahl zwischen **1** und **99**.

- Den Quellen **[base]**, **[updates]**, **[extras]** geben wir die **priority=1**.
- **[centosplus]**, **[contrib]** erhält die **priority=2**.
- Den **3rd party repositories**, wie z.B. **[epel]** oder **[rpmforge]** geben wir nun eine grössere Priorität; also **priority=10**.

Beispiel: EPEL Repository priorisieren & testen (optional)

Zum Schluss, werden noch (Falls der optionale Schritt mit dem priorities gemacht wurde) die Priorität der neu installierten Paket-Quelle angepasst. Dies wird dann so gemacht:

```
# vim /etc/yum.repos.d/epel.repo
```

```
[epel]
name=Extra Packages for Enterprise Linux 7 - $basearch
#baseurl=http://download.fedoraproject.org/pub/epel/7/$basearch
```

```
mirrorlist=https://mirrors.fedoraproject.org/metalink?repo=epel-7&arch=$basearch
failovermethod=priority
enabled=1
# rebermi : 2017-02-06
# default: gpgcheck=0
gpgcheck=1
# default: unset
priority = 10
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-EPEL-7

[epel-debuginfo]
name=Extra Packages for Enterprise Linux 7 - $basearch - Debug
#baseurl=http://download.fedoraproject.org/pub/epel/7/$basearch/debug
mirrorlist=https://mirrors.fedoraproject.org/metalink?repo=epel-debug-7&arch=$basearch
failovermethod=priority
enabled=0
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-EPEL-7
gpgcheck=1

[epel-source]
name=Extra Packages for Enterprise Linux 7 - $basearch - Source
#baseurl=http://download.fedoraproject.org/pub/epel/7/SRPMS
mirrorlist=https://mirrors.fedoraproject.org/metalink?repo=epel-source-7&arch=$basearch
failovermethod=priority
enabled=0
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-EPEL-7
gpgcheck=1
```

Hier wird nun nochmals die schon erwähnte **priority=10** Zeile eingetragen.

Zum Testen des EPEL Repositories, kann einfach nach einem Paket gesucht werden, welchen nur in den epel-repositories existiert. *Zum Beispiel "sks"*

```
# yum info sks
```

```
Available Packages
Name      : sks
Arch      : x86_64
Version   : 1.1.5
Release   : 2.el7
Size      : 768 k
Repo      : epel/x86_64
Summary   : Synchronizing Key Server
```

```
URL      : http://code.google.com/p/sks-keyserver/  
License  : GPLv2+  
Description : SKS is a OpenPGP keyserver whose goal is to provide easy to  
           : deploy, decentralized, and highly reliable synchronization.
```

Wird dieses Paket nun gefunden, so funktioniert auch das Repository einwandfrei.

Last update: **2017/07/10 14:48**