

een RPM maken van een Binary en een systemd unit file

tools installeren

voor deze acties heb je de volgende tools nodig; tar, gzip, rpmbuild. Installeer die door:

```
sudo yum install -y rpm-build redhat-rpm-config rpmdevtools tar gzip
```

directory structuur voor rpmbuild voorbereiden

de rpmbuild tool verwacht een bepaalde directory structuur waarin de files voor de te bouwen rpm zich bevinden, maak deze door:

```
mkdir -p ~/rpmbuild/{BUILD,RPMS,SOURCES,SPECS,SRPMS}
```

rpmbuild configuratie file voorbereiden

rpmbuild haalt een aantal instellingen uit de file .rpmmacros leg de basis hiervoor door:

```
echo '%_topdir %(echo $HOME)/rpmbuild' > ~/.rpmmacros
```

binary en unitfile voorbereiden

files in zip file plaatsen

rpmbuild verwacht een tar.gz file met daarin de directory structuur en files zoals deze op het root filesystem moeten worden geplaatst.

In dit voorbeeld, een simpele agent:

```
mkdir -p ~/appagent-1.0/usr/sbin/  
mkdir -p ~/appagent-1.0/usr/lib/systemd/system/
```

plaats vervolgens de binary en de systemd unit file daar:

```
cp appagent ~/appagent-1.0/usr/sbin/  
cp appagent.service ~/appagent-1.0/usr/lib/systemd/system/
```

zet de permissies zoals die in de eindsituatie moeten zijn:

```
chmod 755 ~/appagent-1.0/usr/sbin/appagent  
chmod 644 ~/appagent-1.0/usr/lib/systemd/system/appagent.service
```

maak er een tarball van:

```
cd ~/
tar -zcvf appagent-1.0.tar.gz ./appagent-1.0
```

en plaats die in de rpmbuild filestructuur:

```
cp appagent-1.0.tar.gz ~/rpmbuild/SOURCES/
```

spec file bouwen

Van het hele proces is dit het meest ingewikkeld, documentatie hierover is vaak gemaakt voor complete compilaties of niet up to date voor de laatste versie van rpm.

maak in de directory ~/rpmbuild/SPECS de file appagent.spec aan met daarin:

```
# Something that need for rpm-4.1 since we have no docs
%define _missing_doc_files_terminate_build 0
%define debug_package %{nil}

Name:          appagent
Version:       1.0
Release:       1%{?dist}
Summary:       An agent for app
License:       Commercial
Source0:       appagent-1.0.tar.gz

Requires(post): info
Requires(preun): info
Requires:       systemd

%description
An agent for the app program.

%prep

%setup -q #unpack tarball

%build

%install
cp -rfa * %{buildroot}

%files
%attr(0755, root, root) "/usr/sbin/appagent"
%attr(0644, root, root) "/usr/lib/systemd/system/appagent.service"
#/*
```

```
%post
#%systemd_post %{name}.service
%systemd_post appagent.service

%preun
%systemd_preun appagent.service

%postun
%systemd_postun_with_restart appagent.service

%changelog
* Wed Jan 05 2022 Pietje Puk <pietje.puk@nergens.nl> 1.0.0
- first draft
```

rpm bouwen

dir doe je simpel door:

```
rpmbuild -ba appagent.spec
```

GPG signing

GPG key aanmaken

Mocht je niet een bestaande GPG key gaan gebruiken kan je er een aanmaken door:

```
gpg --gen-key
```

vul alles in, eventueel ook een passphrase.

controleer door: `gpg --list-keys`

GPG key importeren

kopieer de bestaande GPG-key die je wilt importeren naar je systeem.

importeer de GPG key door: `rpm --import /path/to/YOUR-RPM-GPG-KEY`

controleer door: `gpg --list-keys`

.rpmmacros aanpassen

sign the rpm

Bronnen

<https://wiki.centos.org/HowTos/SetupRpmBuildEnvironment>

<https://blog.packagecloud.io/rpm/rpmbuild/packaging/2015/06/29/building-rpm-packages-with-rpmbuild/>

<https://unix.stackexchange.com/questions/429511/how-to-use-rpmbuild-to-build-a-rpm-package-from-binary-tarball>

<https://fedoraproject.org/wiki/RPMGroups>

<https://stackoverflow.com/questions/5613954/extract-the-spec-file-from-rpm-package>

<https://stackoverflow.com/questions/67094819/why-is-systemd-post-used-in-the-spec-file-not-enabling-or-starting-the-servic>

http://aerostitch.github.io/linux_and_unix/RedHat/build_sample_rpm.html

From:
<https://wiki.auriel.nl/> -

Permanent link:
https://wiki.auriel.nl/doku.php?id=werkinstructies:rpm_van_een_binary&rev=1641392932 

Last update: **2022/01/05 15:28**