

openSUSE:How to detect Tumbleweed?

This tutorial will tell you how to properly detect [openSUSE Tumbleweed](#) when building a package targeting multiple openSUSE releases on [openSUSE Build Service](#). This rule of thumb is an advanced tip that requires some basic knowledge about shell scripting. We assume that you have already achieved that knowledge from somewhere else.

Contents

[\[hide\]](#)

- [1 What? Why?](#)
- [2 Theory](#)
- [3 Examples](#)
 - [3.1 cinnamon-bluetooth](#)
 - [3.2 gpac in packman](#)
- [4 Author](#)

What? Why?

openSUSE has about 4 to 6 releases to support at any time spot, eg. for now, 12.3, 13.1, Tumbleweed, Factory are the mandatory ones, SLE_11_SP3 and SLE_11_SP2 are the "support if you like" ones. There're of course differences between those releases, eg. packages' renaming, packages' versions, which are two most annoying problems we may face in packaging.

Normally, we use the OBS mechanism: `%if %{?suse_version}` to solve the problems. The "suse_version" RPM macro and its variant "sles_version" look like this inside:

```
$ cat /usr/lib/rpm/suse_macros
%suse_version 1310
%sles_version 0
```

They're just hardcode version numbers.

And one fact is that openSUSE developers only bump that version in openSUSE:Factory. openSUSE Tumbleweed cherry-picks new packages like kernel, GNOME, KDE. RPM is not one of them (suse_macros locates in rpm package). That is, openSUSE Tumbleweed has the same "suse_version" but different packages with the latest openSUSE stable release.

Seite 1 / 4

OBS

So the problem came that some packages may need special patches or dependencies for Tumbleweed while the mechanism provided by the Build Service can not achieve the aim.

Theory

1. We link the rpm package ourselves, change the suse_version ourselves, then use:

```
%if %{?suse_version} == "tumbleweed"
```

This method sounds easy that everyone can master, but is complicated in real life:

- You have to submit a new "rpm" to openSUSE Tumbleweed repository, which is really hard to maintain. (Really?)
- Or you have to link rpm for every repository you work on, and those packages can't be submitted to official openSUSE development repositories, because there without the modified rpm package your work will be invalid.

2. We detect "suse_version == 1310" first, then detect specific packages which are different in stable release and Tumbleweed.

Examples

cinnamon-bluetooth

Take "cinnamon-bluetooth" for example:

GNOME upstream removed rfcill functions from gnome-bluetooth in GNOME 3.12, and re-invent the wheels in gnome-settings-daemon. But cinnamon upstream didn't turn their heads to the new codes because Linux Mint is still using GNOME 3.8 even in its latest release. So we have to use two patches provided by other distro packagers to make it build although the result is not functional.

But openSUSE 13.1 is using GNOME 3.10 while Tumbleweed is using GNOME 3.12, so openSUSE 13.1 builds while Tumbleweed not. We have to apply those patches to Tumbleweed only.

So let's check gnome-bluetooth-devel first, which is pkgconfig(gnome-bluetooth-1.0), a dependency for cinnamon-bluetooth.

OBS

```
$ cat %{_libdir}/pkgconfig/gnome-  
bluetooth-1.0.pc | grep Version | sed 's/Version: //'  
3.12.0
```

explanation:

- %{_libdir}/pkgconfig/gnome-bluetooth-1.0.pc queries its pkgconfig file, which always has a version number in it. (You can download gnome-bluetooth-devel package for Tumbleweed and `unrpm` it to see this file) `cat` means to output the whole file
- `grep Version` will return the line containing the `Version` string, eg. Version: 3.12.0
- the `sed 's/<orig>/<replace>/'` will remove 'Version:<whitespace>' to none. so you get 3.12.0

so we can define a macro like this:

```
%define gnome_bluetooth_version %(cat %{_libdir}/pkgconfig/gnome-  
bluetooth-1.0.pc | grep Version | sed 's/Version: //')  
%define is_tumbleweed %(%if %{?suse_version} == 1310 && %{?gnome_bluetooth_version}  
== 3.12.0 ; echo 1 || echo 0)
```

Then use like this:

```
%prep  
%setup -q  
%if 0%{?if_tumbleweed} == 1  
%patch -p1  
%endif
```

gpac in packman

openSUSE frequently renames xulrunner which makes support for them quite hard.

OBS

see:

https://pmbs.links2linux.de/package/view_file/Essentials/gpac/gpac.spec?expand=1&rev=ce1fabced5980e19ac72b3cea1b20897

about how we detect for Tumbleweed. This way we didn't use `cat <pkgconfig file>` but `rpm -q --qf`.

Mehr Informationen unter: https://en.opensuse.org/openSUSE:How_to_detect_Tumbleweed

Eindeutige ID: #1035

Verfasser: Eric Schirra

Letzte Änderung: 2015-03-14 18:03