

# Packaging conventions

GeekosDAW follows some simple packaging conventions to ensure consistency.

These conventions are built on top of the [official openSUSE packaging guidelines](#), which must be fully respected.

## Naming

### Capitals

Capital letters are allowed in packages but generally discouraged, following the official openSUSE guidelines.

### Plugins

Plugins are named like this:

[standard]-[plugin\_name]

[standard] is the lowercase name of the plugin standard (e.g. lv2, vst3, vamp). The correct name for VST2 plugins is vst

[plugin\_name] is the name of the plugin

Some examples of allowed plugin names: ✓ lv2-zyn-fusion ✓ vst3-surge

Some examples of disallowed plugin names: ✗ LV2-zyn-fusion ✗ zyn-fusion-lv2 ✗ FatFrog.lv2

## Versioning

Always use the original versioning scheme of the original program.

When using software without versions, if they are taken from repositories, use the scheme [repository\_type].[branch].[date]

[repository\_type] is the type of the repository (e.g. git, hg, svn, cvs)

[branch] is the name of the branch (e.g. master, trunk, develop)

[date] is the date of the latest commit in the branch in the YYYY.MM.DD (year.month.day) format.

So, assuming the latest commit was pushed to git master on November 18th, 2020:

- Allowed plugin version: ✓ git.master.2020.11.18
- Some examples of disallowed plugin versions: ✗ git.2020.11.18 ✗ git.master.18.11.2020 ✗

git.master  git.master.20201118  20201118

## Patches

Patches are named like this

fix-[issue].patch

Where [issue] is the issue which the patch fixes.

So, assuming a patch was written to fix a segfault by a developer called Rick Deckard on the original repo's "develop" git branch:

- Allowed patch filename: ✓ fix-segfault.patch
- Some examples of disallowed patch filenames:  fix-deckard.patch  fix-git-develop.patch

**All patches that are not openSUSE-specific should be sent upstream.**

**If the patch is taken from the web, please credit the original author and make sure the license allows redistribution.** If it's a patch for a copylefted program, the license is already compliant.

## In-spec patching

Sometimes things can be patched directly from the .spec file, for example by running sed or awk. This leads to packages breaking less often, because patches can rapidly become outdated if the target files are modified by later versions of a program.

For this reason, this practice is allowed (and even encouraged when it concerns things like changing /usr/lib to /usr/%{\_lib}) as long as the changes made are not substantial and don't concern the upstream codebase.

## Static and dynamic linking

Prefer static linking when the original program defaults to it and for the most part, when the packaged dependencies aren't huge. In order to do this, sometimes it's necessary to download some git submodules manually, since OBS can't connect to the Internet.

## .desktop files

As per the official openSUSE packaging policies, all GUI-based packages should have a .desktop file and an icon.

If an official icon does not exist, if you have decent artistic skills try creating one yourself, else use [this placeholder icon](#) (licensed under the GNU GPLv3+).

## RPMLint

All packages should be RPMLint-compliant and strive to have 0 warnings and 0 errors.

Packages with some warnings are allowed.

Disabling some RPMLint directives is allowed as long as:

- The software has been tested by the maintainer
- In the long-term, the issues will be fixed by the maintainer

From:

<http://geekoswiki.tuxfamily.org/> - **geekoswiki**

Permanent link:

[http://geekoswiki.tuxfamily.org/doku.php?id=en:packaging\\_conventions](http://geekoswiki.tuxfamily.org/doku.php?id=en:packaging_conventions)

Last update: **2020/11/19 16:28**

