How to use and handle masked packages

How to use and handle masked packages v0.1, en - LinuxReviews.org

This document explains how to use packages blocked by portage because they are flagged masked, meaning the ebuild for the package in question is new and waits proper testing, has known bugs or for some other reason is considered only to be used by experienced users.

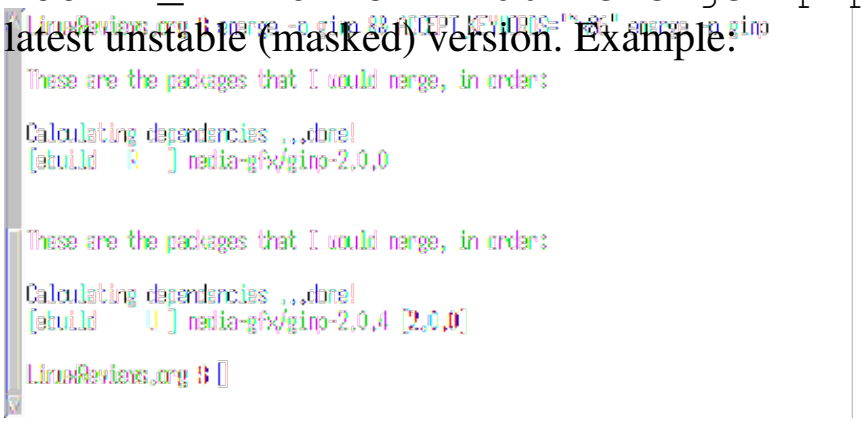
---

1. What is a masked packages, and why are packages masked?

Packages are masked when they are introduced to Gentoo Portage to ensure that the average user gets a stable, usable, working system when using the portage tree. A new version does not need to be better, faster or more stable. The Gentoo developers ensure that upgrading to the latest available versions will not break your system, open security holes or in any other way cause you harm. The masked packages are available to **give you the freedom to choose** between the latest development versions and the stable branches.

1.1. Stable v.s. Unstable

Most packages have several ebuilds available in portage. There is usually a latest **stable version** and a latest **unstable version**. `emerge -p package` will show you latest available stable version and `ACCEPT_KEYWORDS="~x86" emerge -p package` will show latest unstable (masked) version. Example:



1.2. Project has not yet made a stable release

Some ebuilds are for packages still in development and can not be installed without first being un-masked. Example:

```
LinuxReviews.org # emerge gnome-extra/gdesklets-core
Calculating dependencies
!!! all ebuilds that could satisfy "gnome-extra/gdesklets-core" have been masked.
!!! possible candidates are:
- gnome-extra/gdesklets-core-0.26.2 (masked by: ~keyword)
- gnome-extra/gdesklets-core-0.26 (masked by: ~keyword)
- gnome-extra/gdesklets-core-0.26.1 (masked by: ~keyword)

!!! Error calculating dependencies, Please correct.
LinuxReviews.org #
```

1.3. Some packages are "hard" masked for special reasons

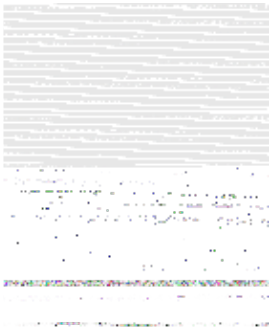
The "hard" masked packages are listed in the file `/usr/portage/profiles/package.mask` (view) and can not be installed without adding a # to comment out the package in question. This is generally a bad idea. The reason why the package is masked is stated next to the package.

```
LinuxReviews.org # emerge -p netscape-communicator

These are the packages that I would merge, in order:

Calculating dependencies
!!! all ebuilds that could satisfy "netscape-communicator" have been masked.
!!! possible candidates are:
- net-www/netscape-communicator-4.80 (masked by: package.mask, ~x keyword)
- net-www/netscape-communicator-4.79-r1 (masked by: package.mask)

!!! Error calculating dependencies, Please correct.
LinuxReviews.org #
```



netscape-communicator is listed in package.mask like this:

```
# agriffis@gentoo.org 18 Aug 2004
# Masking due to security issues that will never be solved since
```

```
# are no new versions forthcoming, bug 56109
net-www/netscape-navigator
net-www/netscape-communicator
```

2. How to install masked packages

You can ask emerge to use the masked version available in portage by setting the variable ACCEPT_KEYWORDS to `ACCEPT_KEYWORDS="~x86"` before running the emerge command. It makes sense to set a command alias in roots `.bashrc` that goes `alias aemerge='ACCEPT_KEYWORDS="~x86" emerge'`.

The alias should be used for checking what version is the latest with `--pretend` (`-p`). Actually installing packages like this is a bad idea because the setting is not stored anywhere, meaning package will be downgraded the next time you do `emerge -u world`.

Gentoo uses the file `/etc/portage/package.keywords` to configure when to use the masked version. Simply list the package and add your arch to unmask packages.

`/etc/portage/package.keywords` can look like this:

```
app-emulation/wine ~x86
net-www/mplayerplug-in   ~x86
media-libs/libquicktime ~x86
net-www/mozilla-firefox ~x86
media-video/transcode  ~x86
```

You can add entries to this file using the echo command:

```
echo app-cdr/k3b ~x86 >>
/etc/portage/package.keywords
```

Explanation: *(I/O Redirection, Bourne Shell Reference)*

```
pgm > file     Output of pgm is redirected to file.
pgm < file     Program pgm reads its input from file.
pgm >> file    Output of pgm is appended to file.
```

2.1. Handy script for unmasking and installing software

Download: gimme

```
#####################################################
# Script to add package names to package.keywords. #
# Originally by Tekmanx, re-written by Hackeron    #
#####################################################

source /sbin/functions.sh

[ "$UID" -ne "0" ] && eerror "Must be root to run this script.
                     exit

[ -z "$1" ] && eerror "Please state a package name (eg. gimme
               exit

[ -d /etc/portage ] || (einfo "Creating /etc/portage directory
                       mkdir /etc/portage )

if [ "$(grep "^$1 ~x86$" /etc/portage/package.keywords)" ]; th
    ewarn "$1 already exists in package.keywords"
else
    einfo Adding $1 to package.keywords and emerging in 5 secc
    sleep 5
    echo $1 ~x86 >> /etc/portage/package.keywords
    emerge $1
fi
```

Install the script and make it executable:

```
cp gimme /usr/local/bin/
chown root:wheel /usr/local/bin/gimme
chmod 555 /usr/local/bin/gimme
```

2.2. How to install hard masked packages

Packages masked by the file `/usr/portage/profiles/package.mask` can be unmasked using the file `/etc/portage/package.unmask` the same way you normally unmask using package.keywords.

You can specify a version using the standard expressions (=, >=, >) or all by only entering app-category/package-name. Both `>=mail-mta/postfix-2.1.1` and `mail-mta/postfix`on a single line will give you the latest postfix.

`/usr/portage/profiles/package.mask` may look like this:

```
>=mail-mta/postfix-2.1.1
sys-apps/portage
```

Do not install masked packages by changing package.mask to make them available. The file is overwritten when you emerge sync.

> Linux Reviews > Gentoo Linux >
How to use and handle masked packages