

# Using custom ebuilds with Gentoo

*LinuxReviews.org*

How to make and use your own ebuilds with Gentoo Linux

---

1. [How to use custom ebuilds](#)
  2. [How to make your own ebuilds](#)
  3. [Using downloaded ebuilds](#)
  4. [Where to look for ebuilds](#)
  5. [Updating ebuilds to new versions](#)
-

# 1. How to use custom ebuilds

Make a folder where you will store your own ebuilds

```
mkdir -p /usr/local/portage
```

and set that folder in `/etc/make.conf`:

```
PORTDIR_OVERLAY=/usr/local/portage
```

The folder structure in your overlay folder should be the same as found in `/usr/portage`.

Use the output from `emerge -s package` or `emerge -pv package` to find its category and name. These reflect actual folders in `/usr/portage`. The ebuild for `sys-apps/portage` is found in `/usr/portage/sys-apps/portage`.

In what folder you place your own ebuilds is up to you, but it needs to fit one of the portage package categories. An ebuild for a new IM program should be placed in `/usr/local/portage/net-im/newprogram/`.

All patches ebuilds use should be placed in a folder `files/` where the ebuild is placed.

The ebuild needs to have a Manifest, a digital signature, for the source files used. These can be made with the command `ebuild`:

```
ebuild myebuild-0.1.1.ebuild digest
```

Once the ebuild is in place and the digital signature is made it can be emerged as usual:

```
emerge myebuild
```

## 2. How to make your own ebuilds

Take a look at the sample skeleton ebuild file founds on all Gentoo systems.

```
less /usr/portage/skel.ebuild
```

With very little editing you can use this file to make your own ebuilds.

- Common Ebuild Writing Mistakes - Ebuild Guidelines  
<http://dev.gentoo.org/~liquidx/ebuildmistakes.html>
- Gentoo Linux Developers HOWTO  
<http://www.gentoo.org/doc/en/gentoo-howto.xml>
- Gentoo Linux Development Policy  
<http://www.gentoo.org/doc/en/policy.xml>
- Contributing Ebuilds  
<http://www.gentoo.org/doc/en/ebuild-submit.xml>

## 3. Using downloaded ebuilds

If you download a plain .ebuild file, then copy the file to an appropriate folder in your portage overlay tree and make a digest for it:

- `mkdir -p /usr/local/portage/app-text/txt2tags`
- `cd /usr/local/portage/app-text/txt2tags`
- `cp /home/you/txt2tags-1.7.ebuild .`
- `ebuild txt2tags-1.7.ebuild digest`

Builds from BreakMyGentoo and others come has archives with a Manifest and digest included. Simply extract these files to `/usr/local/portage/` and emerge them as usual:

- `wget`  
`http://ebuilds.net/gdesklets-core-0.24.1-rc1.tar.gz`
- `tar xfvz gdesklets-core-0.24.1-rc1.tar.gz -C`  
`/usr/local/portage/`

## 4. Where to look for ebuilds

- BreakMyGentoo <http://www.breakmygentoo.net/>
- Gentoo-Portage.com <http://gentoo-portage.com/>
- Latest Portage Additions <http://packages.gentoo.org>

## 5. Updating ebuilds to new versions

If you want to make an ebuild for a new release for slurm, you could:

- `mkdir -p /usr/local/portage/net-analyzer/slurm`
- `cd /usr/local/portage/net-analyzer/slurm`
- `cp -r /usr/portage/net-analyzer/slurm ..`

Then rename the latest ebuild in portage to your new version

- `mv slurm-0.2.3.ebuild slurm-0.3.2.ebuild`

and use that as the basis for your new ebuild. You could also add `-rN` to your ebuild to indicate it is a modified version of some ebuild: `slurm-0.3.2.ebuild.ebuild` becomes `slurm-0.3.2.ebuild-r1.ebuild`.

Now download the source and make a digital signature:

```
ebuild slurm-0.3.2.ebuild digest
```

Now your new ebuild should be available when you do

```
emerge slurm
```

---

Copyright (c) 2000-2004 [Øyvind Sæther](#). Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "[GNU Free Documentation License](#)".

---

> [Linux Reviews](#) > [Gentoo Linux](#) >  
Using custom ebuilds with Gentoo