

원본 출처: [Emerge](#)

emerge

Portage - the heart of Gentoo
emerge - configuration - ebuild repository - dispatch-conf
world file - USE flags - ebuilds - profiles
upgrades - using testing packages - binary packages
tools - gentoolkit - eselect
Portage FAQ - cheat sheet - FAQ

emerge는 포티지에 대한 명령줄 인터페이스이며 대부분의 사용자가 포티지와 상호작용하는 방식입니다. 젠투에서 가장 중요한 명령어 중 하나입니다.

emerge는 젠투 리눅스에서 소프트웨어 패키지를 설치, 업데이트 및 유지 관리하는 데 사용됩니다.



팁

emerge 명령에는 다양한 옵션이 있습니다. 자세한 설명서와 모든 옵션의 전체 목록은 **man emerge**를 참고하세요.



참고 emerge 명령 이외의 포티지 사용법은 [포티지](#) 문서를 참고하세요.

emerge 명령에 대한 몇 가지 일반적인 질문은 [FAQ](#)와 [포티지 FAQ](#)에 답변이 나와 있습니다.

사용법

실행

emerge는 변경될 내용에 대한 풍부한 출력을 제공할 수 있으며, 개별 패키지 또는 시스템에 대한 정보와 경고를 제공합니다. 기본적으로 **emerge** 명령은 요청된 작업을 즉시 수행합니다. **-ask**, **-pretend** 및 **-verbose** 옵션은 포티지가 더 많은 정보를 표시하도록 하는 데 유용합니다.

-help 옵션과 함께 **emerge**를 실행하면 명령줄 옵션에 대한 정보를 제공합니다:

```
user $emerge --help
emerge: command-line interface to the Portage system
Usage:
  emerge [ options ] [ action ] [ ebuild | tbz2 | file | @set | atom ] [
... ]
  emerge [ options ] [ action ] < @system | @world >
  emerge < --sync | --metadata | --info >
```

```

emerge --resume [ --pretend | --ask | --skipfirst ]
emerge --help
Options: -[abBcCdDefgGhjkKlNnOoPqrsStuUvVwW]
          [ --color < y | n > ] [ --columns ]
          [ --complete-graph ] [ --deep ]
          [ --jobs JOBS ] [ --keep-going ] [ --load-average LOAD
]
          [ --newrepo ] [ --newuse ] [ --noconfmem ] [ --nospinner
]
          [ --oneshot ] [ --onlydeps ] [ --quiet-build [ y | n ]
]
          [ --reinstall changed-use ] [ --with-bdeps < y | n >
]
Actions: [ --depclean | --list-sets | --search | --sync | --version
]

```

For more help consult the man page.

아래는 “package”를 설치하는 **emerge** 호출 예제입니다. 옵션(-atv)은 -ask, -tree 및 -verbose에 대한 짧은 옵션입니다. 이 옵션은 계속 진행하기 전에 묻고, 설치할 패키지의 종속성 트리를 표시하고, 출력에 대한 자세한 내용을 표시하도록 **emerge**를 트리거합니다:

```

# emerge -atv package
These are the packages that would be merged, in reverse order:

Calculating dependencies... done!
[ebuild U ] category/package-3.0-r2 [2.0] USE="enabled -disabled
toggled* new% (-unavailable)" MAKE_OPTIONS="-disabled" 777 kB
[ebuild UD ] category/package-2.0 [3.0] 777 kB
[ebuild R ] category/package-1.0 777 kB
[ebuild N ] category/package-0.5 777 kB

Total: 4 packages (1 new, 1 reinstall, 1 upgrade, 1 downgrade), Size of
downloads: 3108 kB

Would you like to merge these packages? [Yes/No]

```

U 기호는 업그레이드될 패키지를, D는 다운그레이드될 패키지를, R은 재설치할 패키지를, N은 새 패키지를 나타냅니다. 대괄호 안에는 이전에 설치된 패키지의 버전이 표시됩니다. 월드 파일에 있는 패키지는 굵게 표시되며, 이는 사용자가 설치한 패키지이고 다른 패키지는 종속성이거나 **시스템 세트**에서 가져온 패키지입니다.



참고 항목

포티지 출력을 읽는 방법에 대한 자세한 설명은 emerge 매뉴얼 페이지의 출력 섹션과 -pretend 옵션을 참조하세요.

포티지에서 “패키지”라는 용어는 “아톰”이라고도 하며, 이 용어는 서로 바뀌서 사용할 수 있습니다.

전 지정을 참조하십시오.

패키지 설치하기

ask 및 verbose 옵션으로 [net-proxy/tinyproxy](#) 패키지를 설치합니다:

```
root #emerge --ask --verbose net-proxy/tinyproxy
```

버전 지정자를 사용하여 설치할 패키지, 버전 및 슬롯, 그리고 어느 이빌드 리포지토리를 지정합니다.

팁



ask(-a) 옵션은 매우 유용하며, 실제 적용을 시작하기 전에 이미지 작업을 검토할 수 있습니다. verbose(-v) 옵션은 포트지가 수행할 작업에 대한 자세한 정보를 표시하며, 종종 유용합니다. 원하는 경우 옵션을 기본값으로 설정할 수 있습니다. 명령줄에서 기본 옵션을 재정의할 수 있습니다(예: ask=n).



경고

ask 옵션을 제공하지 않으면 사용자에게 확인을 요청하지 않고 작업이 수행됩니다. pretend 옵션도 사용할 수 있습니다.

패키지 기능은 설치 시 소프트웨어의 용도에 따라 설정하거나 설정 해제할 수 있는 [USE 플래그](#)에 의해 관리됩니다.

패키지 검색



참고

여기에 설명된 **emerge** 명령의 기본 제공 검색 기능은 다른 도구에 비해 결과가 제한적으로 표시될 수 있습니다. 사용 가능한 최신 버전은 (현재 [profile](#), [make.conf](#), [package.accept_keywords](#) 등에서 파생된) [ACCEPT_KEYWORDS](#)에 의해 제한됩니다. 또한 사용 가능한 최신 버전과 설치된 최신 버전은 슬롯을 인식하지 못하므로 슬롯별로 여러 버전을 표시하지 않습니다. 이로 인해 [eix](#) 또는 [eshowkw](#)와 같은 도구가 **emerge -search**보다 더 많은 결과를 표시할 때 약간의 혼란이 발생할 수 있습니다.

<WRPA center round box 60%> **참고 항목**

패키지를 검색하는 더 고급 방법은 [eix](#), [equery](#) 및 [packages.gentoo.org](#)를 참조하세요.

이름에 프록시가 포함된 패키지를 검색합니다:

```
user $emerge --search proxy
```

이름이나 설명에 프록시가 포함된 패키지를 검색합니다:

```
user $emerge --searchdesc proxy
```

정규식을 사용하여 패키지를 검색합니다:

```
user $emerge -s '%^python$'
```

카테고리의 모든 패키지를 나열합니다:

```
user $emerge -s '@net-ftp'
```

패키지 제거(제거)

좀더 확실한 종속성 파악을 위해 `-depclean(-c)` 옵션을 사용하여 [net-proxy/tinyproxy](#) 패키지를 제거합니다:

```
root #emerge --ask --verbose --depclean net-proxy/tinyproxy
```

이 명령은 현재 설치된 패키지의 종속성으로 필요하지 않은 패키지만 제거합니다.

패키지를 제거할 때 `-depclean`을 사용하는 대신 다음 섹션에 설명된 대로 **emerge -deselect**(또는 **-w** 옵션)를 사용한 다음 종속성이 없는 패키지를 정리하는 방법을 사용할 수 있습니다.



팁

소문자 `-c` 스위치는 `-depclean`의 줄임말이며 안전하지만 시스템 손상의 위험이 있으므로 꼭 필요한 경우에만 사용해야 하는 대문자 `-C` 스위치와 혼동하지 마세요(아래 경고 참조).



경고

`-unmerge (-C)` 옵션은 특별한 이유가 있는 경우가 아니라면 **사용하지 마세요**. 이 옵션을 사용하면 시스템이 작동하는 데 필요한 중요한 패키지가 경고 없이 제거됩니다.

고아 패키지 정리하기



팁

사용하지 않을 가능성이 있는 패키지를 제거하기 위해 `emerge --depclean`을 사용하는 방법에 대한 자세한 내용은 [고아 패키지 제거하기](#)를 참조하세요. [포티지 FAQ](#)도 참조하세요.

패키지 업데이트

패키지를 업데이트하는 방법은 [젠투 업그레이드](#)를 참고하세요.

시스템 정보 얻기

emerge는 문제 해결에 유용한 시스템 정보를 인쇄할 수 있습니다. 이 정보는 [지원](#)을 요청하거나 [버그](#)를 제출할 때 게시해야 하는 경우가 많습니다.

```
user $emerge --info
```

추가 정보는 `--verbose` 플래그를 사용하여 출력할 수 있습니다.

팁

배포 파일 확인 및 (재)다운로드하기

현재 설치된 모든 패키지에 대해 이전에 제거/손상된 배포 파일의 무결성을 다시 확인하고 다시 다운로드하려면 다음과 같이 실행하세요:

```
root #emerge --ask --fetchonly --emptytree @world
```

월드 파일에 종속성을 추가하지 마세요.

종속성을 다시 설치해야 하는 경우 `--oneshot` 옵션을 사용하세요. `emerge` 패키지 명령으로 종속성을 설치하면 [월드 파일](#)에 종속성이 추가되어 문제가 발생할 수 있습니다.

사용자 정의 소스 소프트웨어를 컴파일하기 위해 포티지를 사용하여 종속성을 설치하는 것도 좋지 않으므로 [이벌드를 작성](#)하는 것이 좋습니다.



참고 자료

자세한 내용은 [사용자:샘/포티지 도움말/젠투_시스템 유지](#)를 참고하세요.

이미지 재개하기

여러 패키지의 이미지가 중단된 경우(예: `ctrl+c`, 크래시...) `--resume` 옵션을 사용하여 중단된 패키지에서 이미지를 재개할 수 있습니다. `--keep-going` 및 `--skipfirst` 옵션도 유용할 수 있습니다. 자세한 내용은 이미지 맨 페이지를 참조하세요.

환경 변수를 통해 임시 구성을 포티지에 전달하기



경고

명령줄에서 환경 변수를 포티지에 전달하면 시스템이 일시적으로 변경될 뿐입니다. 이러한 변경 사항은 https://wiki.gentoo.org/wiki/Upgrading_Gentoo 시스템 업데이트를 통해 되돌릴 수 있습니다. 이 방법은 주로 `--pretend` 옵션과 함께 구성 변경의 효과를 미리 볼 때 유용합니다. 포티지 구성 변경은 일반적으로 `/etc/portage/make.conf`, `/etc/portage/package.accept_keywords`, `/etc/portage/package.use` 등과 같은 구성 파일을 통해 이루어집니다.

명령줄에 환경 변수를 선언하여 임시 구성 값을 전달하면 해당 호출에 대한 동작에만 영향을 줄 수 있습니다. 예를 들어, [svg USE 플래그](#)가 활성화된 [app-editors/emacs](#)을 병합하되 이 USE 플래그 설정을 영구적으로 만들지 않으려면 다음과 같이 하세요:

```
root #USE='svg' emerge app-editors/emacs
```

또는 이 빌드에서 `econf` 함수를 사용하는 패키지에 추가 구성 옵션을 전달할 수 있습니다:

```
root #EXTRA_ECONF='--without-compress-install' emerge app-editors/emacs
```

참고



[sudo](#)를 사용하여 `emerge`를 호출하는 경우, `sudo`는 일반적으로 실행되는 환경을 보존하지 않으므로 **sudo** 호출 후에 환경 변수를 설정해야 할 수 있습니다:

```
user $sudo USE='svg' emerge -av app-editors/emacs
```

파일 이미지

파일을 직접 이미지하는 것이 유용한 경우가 있는데, 예를 들어 패키지에 `/usr/lib/libunwind.a`가 필요하지만 어떤 패키지가 이 파일을 제공하는지 알 수 없는 경우를 들 수 있습니다.

```
user $emerge -p /usr/lib/libunwind.a
```

다음은 병합될 패키지의 순서입니다:

종속성 계산... 완료!

종속성 해결에는 2.76초가 걸렸습니다(백트래킹: 0/20).

```
[ebuild R ] sys-libs/llvm-libunwind-17.0.6
```

문제 해결

참고 항목

포티지 문제에 대한 항목은 [사용자:샘/포티지 도움말](#)을 참조하십시오.

'압축 풀기' 단계에서 이미지 패키지 실패

패키지를 압축 해제할 때 다음과 같은 메시지가 나타날 수 있습니다:

```
* Error messages for package dev-libs/libinput-1.16.0:
* The ebuild phase 'unpack' has exited unexpectedly. This type of behavior
* is known to be triggered by things such as failed variable assignments
* (bug #190128) or bad substitution errors (bug #200313). Normally, before
* exiting, bash should have displayed an error message above. If bash did
* not produce an error message above, it's possible that the ebuild has
```

```
* called `exit` when it should have called `die` instead. This behavior
* may also be triggered by a corrupt bash binary or a hardware problem
* such as memory or cpu malfunction. If the problem is not reproducible or
* it appears to occur randomly, then it is likely to be triggered by a
* hardware problem. If you suspect a hardware problem then you should try
* some basic hardware diagnostics such as memtest. Please do not report
* this as a bug unless it is consistently reproducible and you are sure
* that your bash binary and hardware are functioning properly.
```

```
* dev-libs/libinput-1.16.0 패키지에 대한 오류 메시지:
* 이빌드 단계 '언팩'이 예기치 않게 종료되었습니다. 이러한 유형의 동작은
* 변수 할당 실패(버그 #190128) 또는 잘못된 치환 오류(버그 #200313) 등으로 인해
* 트리거되는 것으로 알려져 있습니다. 일반적으로 bash를 종료하기 전에 위의
* 오류 메시지가 표시되어야 합니다. bash가 위의 오류 메시지를 표시하지 않았다면,
* 이빌드가 'exit'를 호출해야 할 때 'die'를 호출했을 가능성이 있습니다.
* 이 동작은 손상된 bash 바이너리 또는 메모리 또는 CPU 오작동과 같은
* 하드웨어 문제로 인해 트리거될 수도 있습니다. 문제를 재현할 수 없거나
* 무작위로 발생하는 것처럼 보인다면 하드웨어 문제로 인해 트리거되었을 가능성이 높습니다.
* 하드웨어 문제가 의심되는 경우 메모스트와 같은 몇 가지
* 기본적인 하드웨어 진단을 시도해 보세요. 지속적으로 재현할 수 있고
* 배시 바이너리와 하드웨어가 제대로 작동하는 것이 확실하지 않은 경우에는 버그로 신고하지 마세요.
```

이 문제는 위의 출력에 나열된 이유 때문일 수도 있지만, 포티지가 이빌드의 소스 파일의 압축을 푸는 데 사용하는 경로의 디스크 공간이 부족해서 발생하는 경우가 많습니다. 이 위치는 `PORTAGE_TMPDIR` 변수를 통해 설정되며, 포티지를 쿼리하여 빠르게 찾을 수 있습니다:

```
user $portageq envvar PORTAGE_TMPDIR
/var/tmp
```

`df` 명령은 `PORTAGE_TMPDIR`이 마운트된 파티션(루트(/) 파티션일 가능성이 높습니다)의 사용 가능한 디스크 공간을 확인하는 데 사용할 수 있습니다. 디스크 공간을 확보하는 방법에 대한 자세한 내용은 [디스크 공간 확보하기](#)를 참조하세요.

참고 항목

[dispatch-conf](#) - 패키지 업데이트 후 구성 파일을 안전하고 편리하게 관리하는 데 사용되는 [포티지](#)에 포함된 유틸리티입니다.

[포티지](#) - 젤투의 공식 패키지 관리자 및 배포 시스템.

From:

<https://gamu.kr/dokuwiki/> - 전자 수첩

Permanent link:

https://gamu.kr/dokuwiki/gentoo/learn_gentoo/emerge

Last update: **2024/05/08 16:07**

