Is this command in archlinux correct - I amtrying to build gimp-git and at the very end after also C compilations, I get errors that dependencies babl and gegl not found but I have already included --nodeps above

The issue you're encountering suggests a mismatch between build-time dependencies and runtime/linking dependencies. Here's what's likely happening:

**The** --nodeps **flag only skips dependency checks during package installation**, but it doesn't help with actual compilation errors. When building GIMP from git, the build process itself needs to find the babl and gegl libraries during compilation and linking.

Here are a few things to check:

1. **Install the development packages**:

bash

sudo pacman -S babl gegl

1. **If you need the latest versions**, you might need the git versions too:

bash

yay -S babl-git gegl-git

1. **Check your build environment**:
   * Make sure pkg-config can find the libraries:

bash

pkg-config --list-all | grep -E "(babl|gegl)"

* Verify library paths are correct:

bash

pkg-config --libs babl gegl

1. **If using an AUR helper**, the correct approach would be:

bash

yay -S gimp-git

This will automatically handle the dependency chain (babl-git → gegl-git → gimp-git).

The --nodeps flag is typically used when you want to force install a package despite missing dependencies, but for building from source, you actually need those dependencies present for the compilation to succeed.

What build method are you using exactly? Are you using makepkg, an AUR helper, or building manually with meson/autotools?

Claude does not have the ability to run the code it generates yet.

D

Can PKGBUILD be modified to skip these dependencies?

**Technically yes, but it's not recommended** because GIMP literally cannot function without babl and gegl - they're core libraries, not optional features.

However, if you want to modify the PKGBUILD, here's what you'd need to change:

## In the PKGBUILD file:

1. **Remove from** depends=() **array**:

bash

*# Remove or comment out these lines*

*# 'babl'*

*# 'gegl'*

1. **Remove from** makedepends=() **array**:

bash

*# Remove these if present*

*# 'babl'*

*# 'gegl'*

1. **Modify the build configuration** to skip these checks:

bash

build() {

*# Add flags to skip dependency checks*

meson setup build \

--buildtype=release \

--prefix=/usr \

-Dforce-fallback-for=babl,gegl \

*# or try:*

*# -Dbabl:required=false \*

*# -Dgegl:required=false*

}

## But here's why this is problematic:

* **GIMP won't compile** without these libraries - they provide essential image processing functions
* **Even if it compiles**, GIMP will crash immediately on startup
* **You'll break the dependency chain** for future updates

## Better alternatives:

1. **If the issue is version conflicts**:

bash

*# Install specific versions*

sudo pacman -S babl=0.1.xx gegl=0.4.xx

1. **If you want to use system libraries instead of git versions**: Edit the PKGBUILD to depend on babl and gegl instead of babl-git and gegl-git
2. **Use a different approach**:

bash

*# Install regular GIMP first, then overlay with git version*

sudo pacman -S gimp

yay -S gimp-git --overwrite

**What specific error are you getting?** The actual error message would help determine if there's a better solution than removing dependencies.

Claude does not have the ability to run the code it generates yet.

D

See the commands. I commented out but since the git was already compiled before again issuing makepkg .... did not help:

[admin@archlinux-nvme gimp-git]$ makepkg -si --nocheck --nodeps ==> Making package: gimp-git 2:3.1.5.r99.3ec63979d8-1 (Tuesday 16 September 2025 03:44:43 PM) ==> WARNING: Skipping dependency checks. ==> Retrieving sources... -> Updating gimp git repo... -> Updating gimp-data git repo... -> Found linux.gpl ==> Validating source files with sha512sums... gimp ... Skipped gimp-data ... Skipped linux.gpl ... Passed ==> Extracting sources... -> Creating working copy of gimp git repo... Reset branch 'makepkg' -> Creating working copy of gimp-data git repo... Reset branch 'makepkg' ==> Starting prepare()... ==> Starting pkgver()... ==> WARNING: A package has already been built, installing existing package... ==> Installing package gimp-git with pacman -U... [sudo] password for admin: loading packages... warning: gimp-git-debug-2:3.1.5.r99.3ec63979d8-1 is up to date -- reinstalling resolving dependencies... warning: cannot resolve "babl-git>=0.1.114", a dependency of "gimp-git" warning: cannot resolve "gegl-git>=0.4.62", a dependency of "gimp-git" :: The following package cannot be upgraded due to unresolvable dependencies: gimp-git :: Do you want to skip the above package for this upgrade? [y/N] y looking for conflicting packages... Packages (1) gimp-git-debug-2:3.1.5.r99.3ec63979d8-1 Total Installed Size: 129.53 MiB Net Upgrade Size: 0.00 MiB :: Proceed with installation? [Y/n] Y

Only debug was built.

The issue is clear now! The package was already built previously, and makepkg is trying to install the **existing package file** which still has the original dependencies baked into it. Even though you commented out the dependencies in the PKGBUILD, the .pkg.tar.xz file in your directory still contains the old dependency information.

## Solution: Force a rebuild

You need to either clean and rebuild, or force makepkg to rebuild:

**Option 1: Clean and rebuild**

bash

makepkg -C *# Clean build files*

rm \*.pkg.tar.xz\* *# Remove old package files*

makepkg -si --nocheck --nodeps

**Option 2: Force rebuild**

bash

makepkg -f -si --nocheck --nodeps

The -f flag forces makepkg to rebuild even if the package file already exists.

**Option 3: Remove just the package files**

bash

rm gimp-git-\*.pkg.tar.xz\*

makepkg -si --nocheck --nodeps

## What happened:

1. You built gimp-git before with the original PKGBUILD (with babl-git/gegl-git dependencies)
2. You modified the PKGBUILD to comment out dependencies
3. makepkg saw the existing .pkg.tar.xz file and said "package already built, let's install it"
4. But that old package file still contained the dependency metadata from the original build
5. pacman -U tried to install it and complained about missing dependencies

After forcing the rebuild, you should see it actually go through the build process again instead of jumping straight to installation.

Claude does not have the ability to run the code it generates yet.

D

That worked! Your Option 2 - Force rebuild. It it would not have then I would be starting on a clea slate - re-downloading git, etc which would have taken significant time.

Excellent! I'm glad Option 2 worked for you. The -f flag is really handy in situations like this - it's much more efficient than starting from scratch since it:

* Keeps the already-downloaded git repositories
* Preserves any intermediate build files that are still valid
* Only rebuilds what's necessary
* Saves you from re-downloading potentially hundreds of MB of source code

This is a common gotcha when modifying PKGBUILDs after an initial build. The key insight is that makepkg checks for existing package files first, and if found, assumes they match the current PKGBUILD - which obviously wasn't the case here after your dependency modifications.

For future reference, other useful makepkg flags for similar situations:

* -C (clean) removes build artifacts but keeps sources
* -c (clean after build) automatically cleans up build files when done
* -r (install build dependencies) if you need to pull in makedepends

The force rebuild approach (-f) is definitely the way to go when you've modified a PKGBUILD but want to avoid the time penalty of re-downloading large git repositories. Smart troubleshooting on your part to try the dependency modification approach!

D

First let me know how to clone pkgconfig and replace gitlab with github.

I think you're asking about how to modify a PKGBUILD that currently uses GitLab as its source to instead use GitHub. Here's how to do that:

## If you want to clone and modify an existing PKGBUILD:

**1. Clone from AUR:**

bash

git clone https://aur.archlinux.org/pkgname.git

cd pkgname

**2. Edit the PKGBUILD to change source URLs:**

bash

nano PKGBUILD

**3. Find and replace the GitLab URL with GitHub:**

bash

*# Change from:*

source=("git+https://gitlab.com/user/repo.git")

*# To:*

source=("git+https://github.com/user/repo.git")

## Example modification:

bash

*# Original GitLab source*

source=("${pkgname}::git+https://gitlab.freedesktop.org/pipewire/pipewire.git")

*# Changed to GitHub mirror*

source=("${pkgname}::git+https://github.com/PipeWire/pipewire.git")

## Update checksums if needed:

bash

updpkgsums *# This updates the checksums automatically*

## Then build:

bash

makepkg -si

**Could you clarify:**

1. What specific package are you trying to modify?
2. Are you looking to switch from GitLab to GitHub for a particular reason (faster downloads, access issues, etc.)?

This will help me give you the exact commands for your specific case.

Claude does not have the ability to run the code it generates yet.

D

But there are sed commands to replace.

You're absolutely right! Using sed commands is much more efficient for bulk replacements. Here are the common sed commands to replace GitLab with GitHub in PKGBUILDs:

## Basic URL replacement:

bash

*# Replace GitLab with GitHub in source URLs*

sed -i 's|gitlab\.com|github.com|g' PKGBUILD

*# Or more specific - replace GitLab freedesktop URLs*

sed -i 's|gitlab\.freedesktop\.org|github.com|g' PKGBUILD

## More targeted replacements:

bash

*# Replace specific GitLab project paths with GitHub equivalents*

sed -i 's|gitlab\.freedesktop\.org/gstreamer|github.com/GStreamer|g' PKGBUILD

sed -i 's|gitlab\.gnome\.org/GNOME|github.com/GNOME|g' PKGBUILD

sed -i 's|gitlab\.com/user/repo|github.com/user/repo|g' PKGBUILD

## Multiple replacements in one command:

bash

sed -i \

-e 's|gitlab\.com|github.com|g' \

-e 's|gitlab\.freedesktop\.org|github.com|g' \

-e 's|gitlab\.gnome\.org/GNOME|github.com/GNOME|g' \

PKGBUILD

## For git clone URLs specifically:

bash

*# Replace git clone URLs*

sed -i 's|git+https://gitlab\.|git+https://github.|g' PKGBUILD

## Preview changes before applying:

bash

*# See what would change (without -i flag)*

sed 's|gitlab\.com|github.com|g' PKGBUILD

## After sed replacements:

bash

updpkgsums *# Update checksums*

makepkg -si

Which specific GitLab → GitHub replacement are you looking to make?

Claude does not have the ability to run the code it generates yet.

D

Got this error at the very end in archlinux. should I install through pacman -Which is this package:

:: unable to satisfy dependency 'libart-lgpl>=2.3.19' required by gimp-git ==> WARNING: Failed to install built package(s).

The libart-lgpl package is **legacy graphics library** from the old GNOME 2 era. It's quite old and has been largely replaced by Cairo and other modern graphics libraries.

## Install the missing dependency:

bash

sudo pacman -S libart-lgpl

## If that doesn't work, try AUR:

bash

yay -S libart-lgpl

## Alternative - check if it's really needed:

Since libart-lgpl is very old, you might want to check if this dependency is actually necessary for modern GIMP. You could try modifying the PKGBUILD to remove this dependency too:

bash

*# Edit PKGBUILD and comment out or remove:*

*# 'libart-lgpl>=2.3.19'*

*# Then rebuild*

makepkg -f -si --nocheck --nodeps

## Why this is happening:

* GIMP-git might have some legacy code paths or optional features that still reference libart-lgpl
* The dependency might be outdated in the PKGBUILD
* Some plugins or legacy file format support might require it

## Quick check:

bash

*# See if the package exists in repos*

pacman -Ss libart

*# Or check AUR*

yay -Ss libart

**Try installing it with pacman first**. If it's not available in the official repos, you might need to get it from AUR, or consider removing this dependency from the PKGBUILD as well if it's truly obsolete.

What does pacman -Ss libart show?