**gimp-git.sh**

162 lines

sh

D

See the attached gimp-git script customised for fedora. It fails even though babl and gegl is built afresh everytime the script is run:

Compiler for C supports arguments -Wunused-function: YES Compiler for C supports arguments -Wunused-value: YES Compiler for C supports arguments -Werror=implicit-function-declaration: YES Library m found: YES Library dl found: YES Found pkg-config: YES (/usr/bin/pkg-config) 2.3.0 Run-time dependency atk found: YES 2.58.0 Dependency babl-0.1 found: NO. Found 0.1.114 but need: '>=0.1.116' Found CMake: /usr/bin/cmake (4.0.0) Run-time dependency babl-0.1 found: NO (tried pkgconfig and cmake) Run-time dependency babl found: NO (tried pkgconfig and cmake) meson.build:418:20: ERROR: Dependency "babl" not found, tried pkgconfig and cmake A full log can be found at /home/admin/gimp-build/gimp/build/meson-logs/meson-log.txt

Edit

The issue is that GIMP requires **babl version 0.1.116 or higher**, but your script is building the latest git version which is **0.1.114**. This seems counterintuitive, but the git master branch might be behind the required version.

Here are the solutions:

**Quick Fix: Checkout a newer babl branch/tag**

gimp-git.sh (Fixed)

Code

#!/bin/bash set -e SKIP\_DEPS=false SKIP\_TESTS=false CLEAN=false while [[ $# -gt 0 ]]; do case $1 in --nodeps) SKIP\_DEPS=true; shift ;; --nocheck) SKIP\_TESTS=true; shift ;; --clean) CLEAN=true; shift ;; \*) shift ;

**Key Changes:**

1. **Removed --depth 1 from babl clone** - This allows fetching tags
2. **Added tag checkout logic** - Fetches all tags and checks out the latest BABL tag (likely BABL\_0\_1\_116 or newer)
3. **Added version verification** - Prints the installed babl version so you can confirm it's >= 0.1.116

**Alternative: If tags don't work**

If the latest tag is still too old, you can manually specify a commit or use the main/master branch of a development version. Check the babl repository to see what the latest stable release is.

Try running the updated script with --clean first to ensure a fresh build:

bash

bash gimp-git.sh --clean

This should resolve the dependency version mismatch!

Retry

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

D

Did not resolve:

ompiler for C supports arguments -Wunused-value: YES Compiler for C supports arguments -Werror=implicit-function-declaration: YES Library m found: YES Library dl found: YES Found pkg-config: YES (/usr/bin/pkg-config) 2.3.0 Run-time dependency atk found: YES 2.58.0 Dependency babl-0.1 found: NO. Found 0.1.114 but need: '>=0.1.116' Found CMake: /usr/bin/cmake (4.0.0) Run-time dependency babl-0.1 found: NO (tried pkgconfig and cmake) Run-time dependency babl found: NO (tried pkgconfig and cmake) meson.build:418:20: ERROR: Dependency "babl" not found, tried pkgconfig and cmake A full log can be found at /home/admin/gimp-build/gimp/build/meson-logs/meson-log.txt

Edit

The issue is that the babl git repository's master branch is actually version 0.1.114, which is **older** than what GIMP needs. We need to check what's actually available. Let me help you investigate and fix this:

babl 0.1.116 release GNOME

10 results

favicon

[Git][archlinux/packaging/packages/babl][main] upgpkg: 0.1.116-1: new upstream release

mail-archive.com



Arch Linux - babl 0.1.116-1 (x86\_64)

archlinux.org



1:0.1.110-1 : babl package : Ubuntu

launchpad.net

favicon

babl/NEWS at master · GNOME/babl

github.com



1:0.1.96-1 : babl package : Ubuntu

launchpad.net



babl-0.1.24 · Tags · GNOME / babl

gnome.org



NEWS · BABL\_0\_1\_96 · GNOME / babl · GitLab

gnome.org



[Gegl-developer] GEGL-0.3.0 babl-0.1.12

gnome.org

favicon

GitHub - GNOME/babl: Read-only mirror of https://gitlab.gnome.org/GNOME/babl

github.com



Cannot compile babl since meson porting (#47) · Issues · GNOME / babl · GitLab

gnome.org

Babl 0.1.116 was just released on October 2, 2025 [[Git][archlinux/packaging/packages/babl][main] upgpkg: 0.1.116-1: new upstream release](https://www.mail-archive.com/arch-commits@lists.archlinux.org/msg894644.html). The issue is that **babl is now hosted on GitLab, not GitHub**. The GitHub repository is just a read-only mirror and may be outdated.

Here's the corrected script that uses the official GitLab repository:

gimp-git.sh (Fixed)

Code ∙ Version 2

#!/bin/bash set -e SKIP\_DEPS=false SKIP\_TESTS=false CLEAN=false while [[ $# -gt 0 ]]; do case $1 in --nodeps) SKIP\_DEPS=true; shift ;; --nocheck) SKIP\_TESTS=true; shift ;; --clean) CLEAN=true; shift ;; \*) shift ;

**Key Changes:**

1. **Changed babl repository** from https://github.com/GNOME/babl.git to **https://gitlab.gnome.org/GNOME/babl.git** (the official source)
2. **Changed gegl repository** to use GitLab as well
3. **Changed gimp repository** to use GitLab
4. **Added version verification** with error exit if babl < 0.1.116
5. Added bc for version comparison (you may need to install it: sudo dnf install bc)

Run with --clean to start fresh:

bash

bash gimp-git.sh --clean

This should now properly fetch babl 0.1.116 from the official GitLab repository!

Retry

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

D

What is this? Again getting the error. I removed gimp-build completely and rerun your revised script. Why it still detects 114 version when 116 is built during script run:

Compiler for C supports arguments -Wsometimes-uninitialized: NO Compiler for C supports arguments -Wtautological-unsigned-enum-zero-compare: NO Compiler for C supports arguments -Wunneeded-internal-declaration: NO Compiler for C supports arguments -Wunused-function: YES Compiler for C supports arguments -Wunused-value: YES Compiler for C supports arguments -Werror=implicit-function-declaration: YES Library m found: YES Library dl found: YES Found pkg-config: YES (/usr/bin/pkg-config) 2.3.0 Run-time dependency atk found: YES 2.58.0 Dependency babl-0.1 found: NO. Found 0.1.114 but need: '>=0.1.116' Found CMake: /usr/bin/cmake (4.0.0) Run-time dependency babl-0.1 found: NO (tried pkgconfig and cmake) Run-time dependency babl found: NO (tried pkgconfig and cmake) meson.build:418:20: ERROR: Dependency "babl" not found, tried pkgconfig and cmake A full log can be found at /home/admin/gimp-build/gimp/build/meson-logs/meson-log.txt

Edit

Ah! The issue is **pkg-config cache**. Even though you're building babl 0.1.116, the system is finding an old 0.1.114 version (probably from a system package). Let me check your script output - did it show "✓ Installed babl version: 0.1.116"?

The problem is likely one of these:

1. **Old system babl package is interfering** - Fedora has babl 0.1.114 installed
2. **pkg-config is finding the wrong .pc file** - searching system paths before /usr/local
3. **ldconfig cache not updating properly**

Here's the fixed script that forces the correct paths and removes system babl:

gimp-git.sh (Fixed)

Code ∙ Version 3

if [[ "$SKIP\_DEPS" == false ]]; then sudo dnf install -y gcc-c++ meson ninja-build pkgconfig gtk3-devel glib2-devel cairo-devel gdk-pixbuf2-devel bubblewrap git json-glib-devel glycin-devel glycin-loaders # Remove system babl/gegl packages th

Now run these commands manually to verify what's happening:

bash

*# Check if system babl is installed*

rpm -qa | grep babl

*# Check what pkg-config finds*

pkg-config --modversion babl-0.1

*# Check what pkg-config finds with correct path*

PKG\_CONFIG\_PATH="/usr/local/lib64/pkgconfig:/usr/local/lib/pkgconfig:$PKG\_CONFIG\_PATH" pkg-config --modversion babl-0.1

*# Check where the .pc files are*

find /usr -name "babl\*.pc" 2>/dev/null

Then run the script again:

bash

bash gimp-git.sh --clean

**If it still fails**, please share the output of those verification commands above so I can see exactly where the old babl is coming from.

Retry

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

D

I ran your fixed script But still error:

ninja: build stopped: subcommand failed. ninja: Entering directory `build' [5/6] Generating gimp-data/images/gimp-splash.png with a custom command (wrapped by meson to set env, to feed input) FAILED: [code=1] gimp-data/images/gimp-splash.png /home/admin/.local/bin/meson --internal exe --unpickle /home/admin/gimp-build/gimp/build/meson-private/meson\_exe\_in-build-gimp.py\_6d994255af837e1bd032a72e5e70f5ccdde7f30d.dat while executing ['/usr/bin/python3', '/home/admin/gimp-build/gimp/tools/in-build-gimp.py', '-nidfs', '../gimp-data/images/gimp-splash.xcf.xz', '--batch-interpreter', 'python-fu-eval', '-b', '-', '--quit'] --- stdout --- GEGL operation missing! GIMP requires the GEGL operation "gegl:npd". This operation cannot be found. Check your GEGL install and ensure it has been compiled with any dependencies required for GIMP. INFO: temporary GIMP configuration directory: /home/admin/gimp-build/gimp/build/.GIMP3-build-config-l0rpkc RUNNING: /home/admin/gimp-build/gimp/build/app/gimp-console-3.1 -nidfs ../gimp-data/images/gimp-splash.xcf.xz --batch-interpreter python-fu-eval -b - --quit Command failed with exit code 1: ['/home/admin/gimp-build/gimp/build/app/gimp-console-3.1', '-nidfs', '../gimp-data/images/gimp-splash.xcf.xz', '--batch-interpreter', 'python-fu-eval', '-b', '-', '--quit'] --- stderr --- GEGL-Message: 14:07:19.682: Module '/usr/local/lib64/gegl-0.4/seamless-clone.so' load error: libgegl-sc-0.4.so: cannot open shared object file: No such file or directory GEGL-Message: 14:07:19.939: Module '/usr/local/lib64/gegl-0.4/npd.so' load error: libgegl-npd-0.4.so: cannot open shared object file: No such file or directory This is a development version of GIMP. Debug messages may appear here. ninja: build stopped: subcommand failed.

Edit

Now we have a different issue! The build got past the babl dependency problem, but GEGL is missing some operations because its shared libraries aren't being found. The problem is LD\_LIBRARY\_PATH isn't set during the build process.

gimp-git.sh (Fixed)

Code ∙ Version 5

echo "Building GEGL..." if [[ ! -d gegl ]]; then git clone --depth 1 https://gitlab.gnome.org/GNOME/gegl.git else cd gegl && git pull && cd .. fi cd gegl rm -rf build meson setup build --prefix=/usr/local ninja-build -C build sudo ninja-build

The key changes:

1. **Created /etc/ld.so.conf.d/usr-local.conf** - This ensures /usr/local/lib64 and /usr/local/lib are in the system library search path permanently
2. **Set LD\_LIBRARY\_PATH during ninja build** - This ensures GEGL can find its own libraries (libgegl-npd-0.4.so and libgegl-sc-0.4.so) during the GIMP build process

Run the script again:

bash

bash gimp-git.sh --clean

Alternatively, you can manually fix the current state without starting over:

bash

*# Configure ldconfig*

echo "/usr/local/lib64" | sudo tee /etc/ld.so.conf.d/usr-local.conf

echo "/usr/local/lib" | sudo tee -a /etc/ld.so.conf.d/usr-local.conf

sudo ldconfig

*# Verify libraries are found*

ldconfig -p | grep gegl

*# Then rebuild GIMP*

cd ~/gimp-build/gimp

rm -rf build

export LD\_LIBRARY\_PATH="/usr/local/lib64:/usr/local/lib:$LD\_LIBRARY\_PATH"

export PKG\_CONFIG\_PATH="/usr/local/lib64/pkgconfig:/usr/local/lib/pkgconfig:/usr/lib64/pkgconfig:$PKG\_CONFIG\_PATH"

meson setup build -Dauto\_features=disabled --prefix=/usr/local

LD\_LIBRARY\_PATH="/usr/local/lib64:/usr/local/lib:$LD\_LIBRARY\_PATH" ninja-build -C build

sudo ninja-build -C build install

This should resolve the missing GEGL operations issue!

Retry

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

[2604/2604] Generating gimp-data/images/gimp-splash.png with a custom command (wrapped by meson to set env, to feed input) FAILED: [code=69] gimp-data/images/gimp-splash.png /usr/bin/meson --internal exe --unpickle /root/gimp-build/gimp/build/meson-private/meson\_exe\_in-build-gimp.py\_f816b9d5dcd858c

pasted

D

Lots of errors at the end:

Edit

The build is progressing further but now has multiple issues:

1. **Missing GObject Introspection typelib files** for Babl and Gegl
2. **Python plugins can't find the typelibs**
3. **Missing python-fu-eval interpreter** (only Script-fu available)

The core issue is that BABL and GEGL need to be built with **GObject Introspection support** enabled. Let me update the script:

gimp-git.sh (Fixed)

Code ∙ Version 7

if [[ "$SKIP\_DEPS" == false ]]; then sudo dnf install -y gcc-c++ meson ninja-build pkgconfig gtk3-devel glib2-devel cairo-devel gdk-pixbuf2-devel bubblewrap git json-glib-devel glycin-devel glycin-loaders gobject-introspection-devel python3-gobje

**Key Changes Made:**

1. **Added gobject-introspection-devel, python3-gobject, and vala** to dependencies
2. **Enabled GObject Introspection in BABL**: -Denable-gir=true
3. **Enabled GObject Introspection in GEGL**: -Dintrospection=enabled
4. **Added GI\_TYPELIB\_PATH** environment variable throughout to help Python find the typelib files
5. **Set GI\_TYPELIB\_PATH in the wrapper script** so it's available when running GIMP

Now run the script with clean:

bash

bash gimp-git.sh --clean

This should build BABL and GEGL with proper introspection support, allowing Python plugins to work correctly. The typelib files (.typelib) will be installed to /usr/local/lib64/girepository-1.0/ and will be discoverable by the Python GI bindings.

Retry

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

The above exception was the direct cause of the following exception: Traceback (most recent call last): File "/root/gimp-build/gimp/build/plug-ins/python/palette-offset/palette-offset.py", line 18, in <module> from gi.repository import Gimp File "/usr/lib64/python3.14/site-packages/gi/impor

pasted

D

Just for your information. In my previous message following errors were not mentioned.

Edit

The typelib files still aren't being found during the build. Let me verify if they were actually created and diagnose the issue:

bash

*# Check if typelib files were created*

find /usr/local -name "\*.typelib" 2>/dev/null

*# Check GEGL build log*

grep -i "introspection" ~/gimp-build/gegl/build/meson-logs/meson-log.txt

*# Check BABL build log*

grep -i "introspection\|gir" ~/gimp-build/babl/build/meson-logs/meson-log.txt

The issue is likely that **introspection wasn't actually enabled** during the BABL/GEGL builds. This can happen if the required dependencies weren't present when meson configured the build.

Let me create a more robust script that verifies introspection is properly enabled:

gimp-git.sh (Fixed)

Code ∙ Version 14

cd babl rm -rf build meson setup build --prefix=/usr/local -Denable-gir=true # Verify introspection will be built echo "Checking BABL build configuration..." if meson configure build | grep -q "enable-gir.\*true"; then echo "✓ BABL GObject intros

Now try the updated script:

bash

bash gimp-git.sh --clean

The script will now:

1. **Verify introspection is enabled** in the meson configuration for both BABL and GEGL
2. **Check that typelib files were actually created** after installation
3. **Exit with an error** if introspection wasn't properly enabled

If it still fails, please share the output from these commands so I can see what's happening:

bash

*# Check if gobject-introspection-devel is actually installed*

rpm -qa | grep gobject-introspection

*# Check what meson saw when configuring GEGL*

grep -A 5 "introspection" ~/gimp-build/gegl/build/meson-logs/meson-log.txt

The core problem is that meson needs to detect gobject-introspection tools during configuration, and if they're missing or not found, it will silently disable introspection support.

Retry

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

D

Not a valid option:

Installed babl version: 0.1.116 (standard\_in) 1: syntax error (standard\_in) 1: syntax error ✓ Babl-0.1.typelib installed successfully Building GEGL... Cloning into 'gegl'... remote: Enumerating objects: 1400, done. remote: Counting objects: 100% (1400/1400), done. remote: Compressing objects: 100% (1217/1217), done. remote: Total 1400 (delta 382), reused 609 (delta 164), pack-reused 0 (from 0) Receiving objects: 100% (1400/1400), 26.81 MiB | 2.26 MiB/s, done. Resolving deltas: 100% (382/382), done. The Meson build system Version: 1.9.1 Source dir: /root/gimp-build/gegl Build dir: /root/gimp-build/gegl/build Build type: native build meson.build:1:0: ERROR: Value "enabled" (of type "string") for option "introspection" is not one of the choices. Possible choices are (as string): "auto", "true", "false". A full log can be found at /root/gimp-build/gegl/build/meson-logs/meson-log.txt

Edit

The meson option syntax is wrong. GEGL uses -Dintrospection=true not enabled. Let me fix that:

gimp-git.sh (Fixed)

Code ∙ Version 18

echo "Building GEGL..." if [[ ! -d gegl ]]; then git clone --depth 1 https://gitlab.gnome.org/GNOME/gegl.git else cd gegl && git pull && cd .. fi cd gegl rm -rf build meson setup build --prefix=/usr/local -Dintrospection=true # Verify intros

Also, let me fix the bc syntax error (missing -l or using wrong syntax):

gimp-git.sh (Fixed)

Code ∙ Version 19

# Verify babl version and typelib BABL\_VERSION=$(pkg-config --modversion babl-0.1) echo "✓ Installed babl version: $BABL\_VERSION" # Simple version comparison without bc MAJOR\_MINOR=$(echo "$BABL\_VERSION" | cut -d. -f1-2) PATCH=$(echo "$BABL\_VERSION"

Now run:

bash

cd ~/gimp-build/gegl

rm -rf build

bash ~/gimp-git.sh

Or start completely fresh:

bash

bash gimp-git.sh --clean

This should now properly enable introspection and create the typelib files needed for Python plugins.

Retry

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

D

I have doubt although gimp build is running it may not succeed: See:

Warning: trying to copy a symlink that points to a file. This currently copies the file by default, but will be changed in a future version of Meson to copy the link instead. Set follow\_symlinks to true to preserve current behavior, or false to copy the link. Installing /root/gimp-build/gegl/bin/lua/gegl\_radial-gradient.lua to /usr/local/share/gegl-0.4/lua Warning: trying to copy a symlink that points to a file. This currently copies the file by default, but will be changed in a future version of Meson to copy the link instead. Set follow\_symlinks to true to preserve current behavior, or false to copy the link. Installing /root/gimp-build/gegl/build/operations/json/grey2.json to /usr/local/lib64/gegl-0.4 Installing /root/gimp-build/gegl/build/operations/json/dropshadow2.json to /usr/local/lib64/gegl-0.4 Installing symlink pointing to libgegl-0.4.so.0.464.1 to /usr/local/lib64/libgegl-0.4.so.0 Installing symlink pointing to libgegl-0.4.so.0 to /usr/local/lib64/libgegl-0.4.so ✓ Gegl-0.4.typelib installed successfully

why gegl showing 0.4? Is that latest?

Edit

Yes, GEGL 0.4 is the current stable API version. The "0.4" refers to the **API/ABI version**, not the release version. The actual release version is shown in the library filename: libgegl-0.4.so.0.464.1 - that's version 0.464.1 (or thereabouts).

This is normal Linux library versioning:

* **0.4** = API version (stable interface)
* **0.464.1** = actual library version (changes with each release)

The typelib is correctly named Gegl-0.4.typelib because Python/GI code imports it as:

python

gi.require\_version('Gegl', '0.4')

So yes, you have the latest git version of GEGL with the 0.4 API, which is what GIMP needs.

**Regarding the build success**: The warnings about symlinks are harmless - those are just meson informational messages about future behavior changes. The build should complete successfully now that:

1. Babl 0.1.116 is installed with introspection
2. Gegl 0.4 is installed with introspection
3. The typelib files exist and are in the right place

The GIMP build should now find the Python interpreter properly. Continue watching the build output - it should get past the splash image generation this time.