

Fixing Ghostscript Issues with dvisvgm on macOS

2025-09-09

This article documents a common issue when rendering TikZ diagrams to SVG on macOS using Quarto and dvisvgm. The problem occurs when Ghostscript is installed but not properly detected by dvisvgm, leading to character jumbling and rendering errors. We provide a complete solution using the `-libgs` option.

blog: <https://tetraquark.vercel.app/posts/dvisvg/>

email: quarktetra@gmail.com

Fixing Ghostscript Issues with dvisvgm on macOS

Your svg's are jumbled?

Line Voltage

The image shows the text 'LINE VOLTAGE' rendered in a serif font. The letters are severely distorted and overlapping, appearing as a dense, illegible mass of black shapes. This is a visual example of the 'jumbled' output mentioned in the text.

Figure 1: An example of a jumbled svg.

Introduction

When rendering TikZ diagrams to SVG using Quarto on macOS, you may encounter issues where Ghostscript is installed but not properly detected by `dvisvgm`. This leads to character jumbling, rendering errors, and warnings about PostScript specials being ignored. This guide provides a complete solution to this common problem.

The Problem

When rendering TikZ diagrams to SVG, you might see warnings like:

```
processing of PostScript specials is disabled (Ghostscript not found)
WARNING: 500 PostScript specials ignored. The resulting SVG might look wrong.
```

Even though Ghostscript is installed on your system, `dvisvgm` cannot find it, resulting in: - Character positioning issues in SVG output - Missing or malformed text in diagrams - Incorrectly sized graphics - PostScript specials being ignored

Diagnosing the Issue

First, let's verify that Ghostscript is installed and where it's located:

```
which gs
# Output example: /opt/homebrew/bin/gs

echo $GS
# Should show the Ghostscript path if set
```

The Solution

The issue is that `dvisvgm` cannot automatically find the Ghostscript library, even when it's installed. The solution is to explicitly tell `dvisvgm` where to find Ghostscript using the `--libgs` option.

Step 1: Find Your Ghostscript Installation

On macOS with Homebrew, Ghostscript is typically installed at: - Intel Macs: `/usr/local/bin/gs`
- Apple Silicon Macs: `/opt/homebrew/bin/gs`

Step 2: Update Your Quarto TikZ Configuration

In your Quarto document, update the TikZ engine options to include the `--libgs` parameter:

```
#| engine-opts:  
#|   template: "../../sharedContent/latexfiles/pgfSimple2.tex"  
#|   dvisvgm.opts: "--no-fonts --bbox=min --libgs=/opt/homebrew/bin/gs"
```

Step 3: Alternative - Set Environment Variable

You can also set the GS environment variable before running Quarto:

```
export GS="/opt/homebrew/bin/gs"  
quarto render your-document.qmd --to html
```

Or add it permanently to your shell configuration:

```
echo 'export GS="/opt/homebrew/bin/gs"' >> ~/.zshrc  
source ~/.zshrc
```

Complete Example

Here's a complete example of a TikZ diagram with proper Ghostscript configuration:

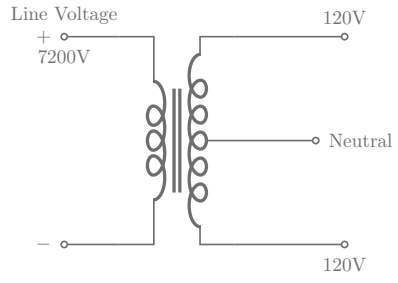


Figure 2: Example TikZ diagram with proper Ghostscript configuration

Before and After Comparison

Before Fix (Without `-libgs`)

When Ghostscript is not properly detected, you'll see: - Warnings about PostScript specials being ignored - Character positioning issues - Incorrectly sized graphics - Text appearing jumbled or misaligned

After Fix (With `-libgs`)

With the proper `--libgs` configuration: - No Ghostscript warnings - Proper character positioning - Correctly sized graphics - Clean, readable text and symbols

Troubleshooting

Issue: Still Getting Ghostscript Warnings

If you're still seeing warnings after adding `--libgs`, check:

1. **Verify the Ghostscript path is correct:**

```
ls -la /opt/homebrew/bin/gs
# Should show the executable exists
```

2. **Check dvisvgm version:**

```
dvisvgm --version
# Should show version 2.6 or later
```

3. **Test dvisvgm directly:**

```
dvisvgm --help | grep -i gs
# Should show --libgs option
```

Issue: Different Ghostscript Installation Path

If Ghostscript is installed elsewhere, find it with:

```
find /usr -name "gs" 2>/dev/null
find /opt -name "gs" 2>/dev/null
```

Then update the `--libgs` path accordingly.

Issue: Permission Errors

If you get permission errors, ensure the Ghostscript executable is readable:

```
chmod +r /opt/homebrew/bin/gs
```

Alternative Solutions

Method 1: Environment Variable (Recommended for Global Fix)

Add to your shell configuration file:

```
# For zsh (default on macOS)
echo 'export GS="/opt/homebrew/bin/gs"' >> ~/.zshrc
source ~/.zshrc

# For bash
echo 'export GS="/opt/homebrew/bin/gs"' >> ~/.bash_profile
source ~/.bash_profile
```

Method 2: Quarto Project Configuration

Create a `_quarto.yml` file in your project root:

```
format:
  html:
    execute:
      engine-opts:
        tikz:
          dvisvgm.opts: "--no-fonts --bbox=min --libgs=/opt/homebrew/bin/gs"
```

Method 3: System-wide dvisvgm Configuration

Create a dvisvgm configuration file:

```
mkdir -p ~/.texlive/texmf-var/tex/latex/dvisvgm
echo '--libgs=/opt/homebrew/bin/gs' > ~/.texlive/texmf-var/tex/latex/dvisvgm/dvisvgm.cfg
```

Summary

The key to fixing Ghostscript issues with `dvisvgm` on macOS is to explicitly specify the Ghostscript library path using the `--libgs` option. This ensures that:

1. **PostScript specials are processed correctly** - No more warnings about ignored PostScript specials
2. **Character positioning is accurate** - Text and symbols appear in the correct locations
3. **Graphics are properly sized** - SVG output matches the intended dimensions
4. **Rendering is consistent** - No more character jumbling or misalignment

The most reliable approach is to add `--libgs=/opt/homebrew/bin/gs` (or the appropriate path for your system) to the `dvisvgm.opts` in your TikZ engine configuration.

Conclusion

This solution resolves the common issue where Ghostscript is installed but not detected by `dvisvgm`, ensuring that TikZ diagrams render correctly to SVG in Quarto documents on macOS. The fix is simple but crucial for proper diagram rendering.