INKSCAPE®
Draw Freely.

Automating Inkscape

Ted Gould
Libregraphics @ SCALE 23x
March 8, 2026

WHOAMI

- Original Developers of Inkscape
  - Forked of Sodipodi with Bryce, Mental and Nathan
  - Developed extensions system
  - Presented at SCALE 5x
- **Currently**: Inkscape PLC
- **Professionally**: Building a GPU security startup

Think differently about graphics

- **Traditionally**: Every graphic design was a manual process in a GUI.
- **Today**: Graphics need to go everywhere and be flexible.
- **Use Text**: Scalable Vector Graphics (SVG) are fundamentally XML.
- **Solution**: Programmatic manipulation of XML + The Inkscape Rendering Engine.

Architectural Evolution: 0.92 vs. 1.x

- **Legacy (0.92.x):** Required **`--without-gui`** flag.
- **Modern (1.x):** Headless by default for exports/queries.
- **POSIX Alignment:** Dropped **`-f`** (file) flag; input is now a positional argument.
- **Consolidation:** Unified export structure under **`--export-filename`**.

From "Verbs" to "Actions"

- **Verbs (Legacy):** Deeply coupled to GUI; required display servers (X11).
- **Actions (Modern):** Atomic, headless operations interacting with the DOM.
- **Chaining:** Multiple operations in one command via semicolons.
- **Syntax:** `--actions="[id]; [id]"` replaces limited GUI-dependent hooks.

Verbs were essentially menu-click simulators. Actions are true API calls. Use `inkscape --action-list` to see them all.

INKSCAPE® Draw Freely.

The "Hello World" of Automation

- Input: A basic SVG file (e.g., **logo.svg**).
- Goal: Export a clean PNG from a vector source.

This is the most common entry point. Note that the output filename is explicit; if omitted, Inkscape defaults to the input name with the new extension or last export.

Speed Optimization: Shell Mode

- Problem: Restarting Inkscape for 1,000 files is slow
- Solution: **`--shell`** mode a REPL
- Benefits
  - Engine remains resident in RAM
  - Scripts pipe commands continuously into one process

History: OpenStreetMap Osmarender

- Pipeline: **OSM XML** → **XSLT** → **SVG** → **PNG**.
- The Role of Inkscape
  - Used as a distributed "printer" for tiles.
  - Unique support for text-on-path labels and SVG filters.

Metatiling & Slicing (The OSM Way)

- Input: A massive map SVG (**map.svg**) representing a 4×4 tile grid.
- Goal: Slice one specific 256×256 tile from a larger canvas.

Software: Multi-Resolution Icons

- Input: A single high-detail master logo SVG.
- Goal: Generate a standard 32px icon and a high-DPI 64px version.

NOTE: You probably want several size 'ranges' with different detail

Game Dev: Procedural Spritesheets (1/2)

- Input: One SVG with frames on layers named **walk1**, **walk2**, etc.
- Output
  - Export a specific frame by its internal XML ID.
  - Combine into an animation with ImageMagick

Game Dev: Procedural Spritesheets (2/2)

INKSCAPE®
Draw Freely.

Data Injection (1/2)

- Input: A template SVG with placeholders like **%VAR_name%**.
- Output: Batch-produce personalized ID cards or conference badges.

Data Injection (2/2)

Scientific Poster

- Input: A poster template with a placeholder for a plot.
- Goal: Embed a new data plot into a vector layout headlessly.

Hardware: Robotic Pen Plotters

- Input: Generative art with thousands of overlapping lines.
- Goal: "Clean" the SVG by converting all text and objects to paths.

Inkstich

- Input: A simple logo SVG.
- Goal: Add hardware "Trim" commands to specific paths before exporting.

The Ink/Stitch CLI allows for "Stitch Plans" to be generated headlessly for quality control in manufacturing.

INKSCAPE®
Draw Freely.

Fabrication: CNC & Laser Cutting

- Input: A CAD design for a mechanical part.
- Goal: Offset the path (outset) to compensate for the laser beam width (kerf).

Security in Headless Environments

- **Risk**: Malicious XML External Entities (XXE) in user-provided SVGs.
- **Mitigation**: Use a pre-processor like **svg-hush**.

Summary

- **The Engine**: Inkscape is now a CLI powerhouse, not just a drawing tool.
- **Growing**: Now with actions and the shell automation is stable and GUI-free.
- **Takeaway**: Your SVGs can effectively be used for programmatic consumption.