Your tools and the runner's tools

This page answers: *which versions of Magic, KLayout and the PDK check your layout on GitHub, how do they differ from the workbench's, and why does the course install a second set?*

What a cloud runner is explains that a runner installs its tools fresh for every job, at whatever versions the workflow asks for. In an analog project, the tools that decide whether your layout passes are precheck's, and their versions aren't the workbench's.

Three sets of tools

Your GDS goes through three sets of layout tools on its way from your keyboard to a green run:

Where	Magic	KLayout	sky130A PDK	Used for
The workbench (`hpretl/iic-osic-tools:2026.08`)	8.3.681	0.30.11	`026824c7`	Whatever you run in it yourself, unless you choose otherwise
The `gds` job's `custom_gds` step	–	0.29.12	–	The OASIS copy and the PNG render; it never checks anything
The `precheck` job	8.3.568	0.30.8	`0536d02d`	All 15 checks

The precheck versions are fixed in two places. The PDK commit is written into the `precheck` action at the tag your workflows use, `ttsky26d`. Magic and KLayout come from a Nix file in Tiny Tapeout's support tools, which pins each to a commit of the Nix package collection, and a precheck log shows what that gave: the job prints `Magic 8.3 revision 568`, and the KLayout it runs is from a folder called `klayout-0.30.8`. The support tools themselves aren't pinned at all: the action checks out their `main` branch on every run, so precheck's own code can change between two runs of the same commit. In November 2026, `main` was at `9d13ff3`, and precheck's code hadn't changed since Oct 29.

Why the versions matter

Tiny Tapeout's analog specs say "Make sure the version of Magic matches the PDK", and the reason is that the design rules live in the PDK's technology files and Magic reads them. A different Magic, or the same Magic with a different PDK, is in principle a different check.

In practice the differences are smaller than that sounds. When the course's tools were set up, we ran precheck on 11 commits from 6 public analog projects whose runs on GitHub were still readable: six commits that had failed precheck and five that had passed. We ran each one twice, once with the runner's versions and once with the workbench's. With the runner's versions, the table matched the runner's every time, row for row and message for message; for two of the failing runs, the Magic report was the same file as the one the runner uploaded, byte for byte. The workbench's versions gave the same pass or fail on all 11, though some counts along the way differed: on one layout, KLayout 0.30.11 counted 7,089 edge pairs at one rule of the deck where 0.30.8 and the runner counted 7,053, and 1,303 where they counted 1,258, and both still ended at the same 1,740 violations. Only one of the 11 layouts had a Magic DRC error, which is too few to say that Magic 8.3.681 always agrees with 8.3.568. The technology files of the two PDK commits differ by one line in their `wiring` section and in the version string; the DRC section is identical.

So the course treats precheck with the runner's versions as the check, and the workbench's tools as a second opinion.

What the course installs

In Lab 01, Part 1, `make install` puts precheck's versions next to the workbench's, in `labs/common/tt-tools/.local` of the course folder, without changing the image: Python 3.11 with precheck's packages at the runner's versions, the PDK at `0536d02d` through the same `ciel` tool the runner uses, KLayout 0.30.8, Magic 8.3.568 built from its source, and the support tools at `9d13ff3`. It takes about a minute and a half and 1.8 GB. `make check-tools` prints each version next to the runner's and the workbench's.

A few things are different from the runner, none of which the comparison above found to matter:

- Python 3.11.16 here, 3.11.18 on the runner, because the workbench's installer offers nothing newer in 3.11. The packages are the same versions, built for 3.11.
- Magic is compiled here without its 3D display, because the image has no headers for it.
- KLayout is klayout.org's own package of 0.30.8, not Nix's.
- The pip `klayout` module that precheck imports is pinned at 0.30.13. The runner takes whatever 0.30.x is newest on the day, which was 0.30.12 on Nov 2 and 0.30.13 on Nov 15.
- The support tools are pinned at `9d13ff3`, where the runner follows `main`.

Which tools each of your `make` targets uses:

Target	Magic	KLayout	PDK
`make` (build the layout)	8.3.568	–	`0536d02d`
`make precheck`	8.3.568	0.30.8	`0536d02d`
`make lvs`	8.3.568 to extract, the workbench's netgen to compare	–	`0536d02d`
`make drc`	the workbench's 8.3.681	–	the workbench's `026824c7`
`make view`	the workbench's 8.3.681, with a display	–	the workbench's

`make drc` is there so you can compare the workbench's Magic and PDK with precheck's. On the layouts we tried it on while writing the course, it agreed with precheck's Magic DRC row about whether there were errors: 0 on Lab 01's frame, on Lab 03's finished inverter and on the empty sandbox, and 8 on a Lab 03 wire with its metal 2 moved off its via, where precheck's row was ❌.

The course builds your layout with precheck's Magic, rather than the workbench's, for two reasons. It's the version precheck checks with, which is what the specs' "matches the PDK" asks for. And the workbench's Magic 8.3.681 sometimes crashes when it runs a script in batch mode. When we set up the course's tools, we saw it crash on exit after it had written its files, and while this course was being written it crashed once halfway through writing a LEF and left the file 222 lines short. So every `make` target judges a Magic run by the files it wrote and never by its exit status, and you'll need the same habit in Lab 05.

When Tiny Tapeout moves the tag

All of this was read from the tag `ttsky26d`, and it can change: each new shuttle brings a new template with a new tag, and Tiny Tapeout sometimes moves an existing tag to a new commit, as Reading a workflow file describes. Your repository keeps the tag it was made with, so its runs keep the same PDK commit, though the support tools still follow `main`. The course's pins live in one file, `labs/common/tt-tools/pins.env`, with the steps for reading them again at its top, and whoever maintains the course reads them again for each new tag.

Check yourself

1. Your layout passes `make drc` with 0 errors and fails precheck's Magic DRC row on GitHub. Which do you believe, and what would you run next?

   Answer: A1, in Answers at the end.

2. Two runs of the same commit, a month apart, give different precheck results. Your files didn't change. Name two things that could have.

   Answer: A2, in Answers at the end.

Answers

A1. Check yourself

Precheck's: it's the check that counts, with its own Magic and PDK. Run `make precheck`, which uses the same versions, and read `magic_drc.txt` in `tt_submission/precheck_reports/` for the rule and the place. If `make precheck` passes too, look at what differs between your machine and the run: most often, the GDS you committed isn't the one you checked.

A2. Check yourself

The support tools' `main` branch, which the action checks out on every run, and the tag itself, if Tiny Tapeout moved it. The log records both: the commit the tag pointed to, in the line that downloads the action, and the support tools' commit in the checkout step.