Magic cheat sheet

This page lists every Magic command the course uses. You can type any of them into the `tkcon` console, and where a table has a keyboard column, that column gives the key or mouse action that does the same thing in the layout window.

Starting

```
cd <a directory with a .magicrc in it>
magic -d X11 <cellname>          # opens layout1 + tkcon
magic -dnull -noconsole script.tcl   # batch, which is what the labs do
```

`-d X11` selects the X11 display driver, which is the one that works in this container. The `-d XR` driver can fail with a `BadAlloc` error and leave you looking at a black window, so stick with X11.

The image's `~/.magicrc` loads whichever PDK `$PDK` names. Magic also reads a `.magicrc` from the directory you start it in, which is why you `cd` first. Each lab package has one of these, and it loads SKY130 only if nothing has loaded it yet:

```
if {[tech name] eq "sky130A"} {
    # Already loaded by ~/.magicrc. Nothing to do.
} elseif {[info exists env(PDK_ROOT)]} {
    source $env(PDK_ROOT)/sky130A/libs.tech/magic/sky130A.magicrc
} else {
    puts stderr "PDK_ROOT is not set. Try:  export PDK_ROOT=/foss/pdks"
}
```

The box

Console	Keyboard / mouse	
`box <x1>um <y1>um <x2>um <y2>um`	left-click, then right-click	set the box
`box values`		read it back, **in units of 0.005 µm**
`box width`, `box height`		read the box's width or height
`findbox zoom`		zoom the view to the box
	`z` / `Z`	zoom in / out
	`v`	view everything

Always write `um`. Magic reads a number with no unit as lambda, which is 0.01 µm here, and it doesn't warn you when it does.

Painting

Console	Keyboard	
`paint <layer>`	middle-click on a layer swatch	fill the box
`erase <layer>`	middle-click on empty space	clear the box
`erase`		clear every layer in the box
`copy` / `move`		copy or move the current selection

Console	Keyboard	
`select`	`s`	select what is under the cursor
`select area`	`a`	select everything in the box
`select top cell`		select the whole cell
`undo` / `redo`	`u` / `U`	
`tech layers`		print every layer name and alias

Layers used in this course

```
nwell pwell            ndiff pdiff           ntap ptap
ndc pdc                ntapc ptapc           pc
poly                   li                    mcon
m1 m2                  via1                  pfethvt
```

`ndc`, `pdc`, `pc`, `ntapc`, `ptapc` and `mcon` are **contact types**, which means that painting one fills the whole stack (the lower layer, the cut and the upper layer) in one step. For a single contact, the cut has to be exactly 0.17 µm square.

Labels and ports

```
box 1.30um 2.40um 1.55um 2.65um
label A c li           ;# name the geometry under the box, on layer li
select top cell
port makeall           ;# turn every label in the cell into a terminal
port <name> index <n>  ;# set a port's number, if order matters to you
```

A label always goes into the cell you're editing, and it names a net only if it sits on geometry **belonging to that cell**. If you place one over a shape that belongs to a subcell, Magic moves it onto empty space (`Moving label "A" from locali to space`), where it names nothing, so paint a small shape of your own over the subcell's and label that.

DRC

```
drc check              ;# re-run
drc catchup            ;# wait for it to finish (batch scripts need this)
drc count total        ;# how many violations: the number you want
drc count              ;# careful: prints "Cell X has 3 error tiles." and then
                       ;#   "Total DRC errors found: 0" - that second line is a
                       ;#   hierarchical total and reads 0 on a flat cell
drc why                ;# name the rules broken under the box (set one first:
                       ;#   with no box, 'select top cell')
drc listall why        ;# every violation in the cell, with rule text
drc euclidean on       ;# measure corners properly rather than as squares
drc style drc(full)    ;# the rule set that catches missing well taps
drc style drc(fast)    ;# back to the default
```

The `Drc` menu on the layout window does the same things with the mouse:

Menu	Same as
Drc → DRC update	`drc check`. Use it when a cell you've just opened shows a green ☑ DRC and no hatching.
Drc → DRC report	`drc listall why`. It doesn't need a cursor box.
Drc → DRC Find next error	Puts the box on the next violating tile *and zooms to it*, which makes it the fastest way to find a hatched area you can't see at the current zoom. It has no console equivalent worth typing.
Drc → DRC Complete / DRC Fast	`drc style drc(full)` / `drc style drc(fast)`

Magic doesn't check a cell you have only opened. Its background checker re-checks the tiles you *edit*, and loading a `.mag` from disk doesn't edit anything, so a cell with real violations opens showing a green ☑ DRC and no white hatching. Run `drc check` (or `Drc → DRC update`) every time you open a cell, before you believe what the DRC indicator says.

Extraction

```
extract do local       ;# this cell
extract all            ;# and every subcell
ext2spice lvs          ;# preset: devices only, hierarchical, for netgen
ext2spice -o out.spice
ext2spice cthresh 0    ;# keep every parasitic capacitance
ext2spice cthresh infinite   ;# back to none
```

Hierarchy

```
addpath ../mag         ;# where to find cells (set it AFTER any cd)
box 3um 0um 3um 0um
getcell my_inv         ;# place an instance with its origin at the box
identify inv2          ;# name this instance
expand                 ;# draw subcells properly instead of as outlines
flatten -nolabels flat ;# copy the whole hierarchy into one cell
cellname list children ;# what is instanced in this cell
```

Files

```
load <cell>            ;# read cell.mag and make it the edit cell
load <cell> -force     ;# start a new empty cell with that name
save <cell>            ;# write cell.mag
cd <dir>               ;# Magic has its own working directory
gds write out.gds      ;# the file a foundry would take
property FIXED_BBOX    ;# a cell's placement outline, if it has one
```

Batch scripts

```
cd results
addpath ../mag
load my_inv
```

```
select top cell
drc euclidean on
drc check ; drc catchup
puts "DRC [drc list count total]"
extract do local ; extract all
ext2spice lvs
ext2spice -o my_inv.spice
quit -noprompt
```

The lab scripts follow this pattern, and you run one with `magic -dnull -noconsole thatfile.tcl`. Put `cd` **before** `addpath`, because `addpath` stores the path exactly as you typed it and doesn't re-resolve it, so a relative path like `../mag` given before a `cd` points at the wrong place afterwards.

netgen (LVS)

```
netgen -batch lvs \
  "layout.spice cellname" \
  "schematic.spice cellname" \
  "$PDK_ROOT/sky130A/libs.tech/netgen/sky130A_setup.tcl" \
  report.txt
```

Look for `Circuits match uniquely.`, which means the layout and the schematic agree. If you get anything else, read the report from the bottom up.

KLayout (viewing)

```
klayout -nn $PDK_ROOT/sky130A/libs.tech/klayout/tech/sky130A.lyt out.gds
```

`-nn` loads the SKY130 technology, so the layers are named and coloured properly.