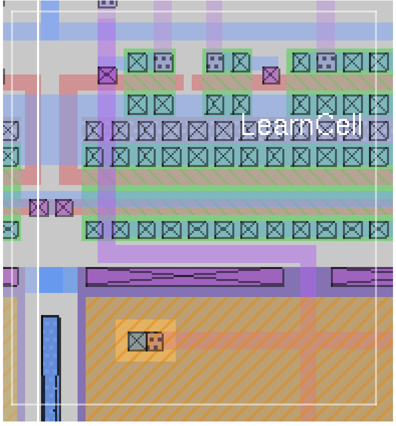
Magic VLSI Layout Tool Version 8.3



Magic User's Guide

Table of Contents

Command-line invocation (usage)
Script Invocation
Magic command summary
Magic tutorials

Obligatory Screenshot

Screenshot of Magic

> This screenshot, from Magic version 7.2, shows off a number of features of the Tcl version, including the cell manager window, the tech manager window, the toolbar, the console command-line entry window, and popup dialog boxes. Also shows off the version 7.1+ features of the OpenGL display and the non-Manhattan geometry extension.

Magic version 8.3 Usage (command-line invocation)

Basic usage:

magic [**-noc**[**onsole**]] [**-now**[**rapper**]] [**-norc**[**file**]] [**-d** *devType*] [**-T** *technology*] [**-rcfile** *startupFile*] [*file*]

where:

-noconsole
: (Tcl version only) Uses the calling terminal for terminal-based command-line input. Otherwise, a Tk console window is used.

-norcfile

(all versions from 7.5.56) Don't read in the user's home "`.magicrc`" file and the current directory "`.magicrc`" files on startup. The site-wide file will still be sourced.

-nowrapper

(Tcl version only) Magic layout windows use the GUI wrapper, including cell and technology manager windows, layer toolbar, and file menu.

-d *devType*

(all versions) Select the graphics interface at runtime. Specifying an invalid *devType* will result in a list of known types. The possible values of *devType* are determined at compile time, but the usual ones are `NULL` (no graphics), `X11`, and `OPENGL`. Magic version 8.2 introduced the additional interface `CAIRO`. "`OGL`" and "`XR`" are common shorthand for the OpenGL and Cairo interfaces. `X11` is the usual default.

-T *technology*

(all versions) Select the appropriate technology (`.tech`) file. At present (this is on the to-do list), magic cannot change technology after startup. So the technology file corresponding to the layout to be loaded must be supplied to the command line at startup. The default technology is `scmos`, which is included with the magic source distribution. The complete list of available technology files depends on what has been installed on the system (see the technology file page for details).

-rcfile *startupFile*

(all versions from 7.5.56) Source the contents of file *startupFile* instead of the default file named "`.magicrc`" in the user's home directory and in the current working directory.

NOTE: The startup file is not expected to generate a layout and execute layout-related commands. However, this can be done by putting the "openwrapper" command in the script before all layout-related commands. In versions of magic starting with 8.1.31, magic will detect that a layout window exists after running the startup script, and not attempt to create another one.

file

(all versions) Load the layout (`.mag`) file *file* into the layout window on startup.
(versions from 8.1.102) Load the file *file*, which can be a layout file, a CIF or GDS file, a LEF file, or a TCL script. There can be any number of files on the command line, separated by spaces.
(versions from 8.3.185) The behavior is the same as above, but if any TCL script is passed on the command line, then all arguments after the name of the TCL script are assumed to be arguments passed to the script and not input files to be processed by magic. The TCL script can see the whole command line argument list in the variable `$argv` (length `$argc`).

Other standard usage:

magic [**--version|--commit|--prefix**]

magic [**-dnull**] [*file*]

magic [**-r**[**ecover**]]

where options are as follows:

`--version`

> This option reports the full version number of magic (*major.minor.revision*) to the terminal and quits (**-dnull -noconsole** implied).

`--commit`

> This option reports the full git commit number of magic to the terminal and quits (**-dnull -noconsole** implied).

`--prefix`

> This option reports the full path of the installed magic run-time libraries (default "`/usr/local/lib`") to the terminal and quits (**-dnull -noconsole** implied).

`-recover`

> This option recovers a layout after a crash. Note that crash recovery files are only *automatically* created and updated by the Tcl/Tk version of magic. A single file containing multiple layouts is placed in the `/tmp` directory. Upon normal program exit, it is removed. However, if magic terminates abnormally due to a program bug, reception of a termination signal from the operating system, or a system crash or shutdown, the file will remain and can be recovered. It is *very* important that you recover the file from the same directory where it was initially created; otherwise, if startup conditions are different (such as a different technology specified), layout may be lost.

`-dnull` *file*

> This option starts magic without graphics. It is appropriate for running magic in batch mode from a script. Note that there is a subtle difference between options "`-d null`" and "`-dnull`". The former starts magic without the ability to create a layout window, but still invokes the graphics initialization routines (in the Tcl/Tk version, a Tk window may briefly appear). The latter

form specifically ignores all graphics and therefore runs with less overhead on startup.

Complete usage information:

magic [**-noc**[**onsole**]] [**-now**[**rapper**]] [**-nowindow**] [**-d** *devType*] [**-T** *technology*] [**-m** *monType*] [**-D**] [*file*]

where the additional options not covered above are:

`-nowindow`

(Tcl version only) Run without displaying an initial layout window. This is used mainly for GUI wrapper scripts which like to generate and handle their own windows.

`-m` *monType*

(obscure) *monType* names a monitor type. This is used in the search for the colomap file name, which is designated `<tech>.<planes>.<mon>.cmap1`. The default is "`std`" (corresponding to colormap file "`mos.7bit.std.cmap1`". The only other monitor type for which colormaps exist in the distribution is "`mraster`". This provides a way for users to override the system color assignments.

`-D`

(all versions) Run in Debug mode.

Obsolete usage information:

magic [`-g` *gPort*] [`-i` *tabletPort*] [`-F` *objFile saveFile*] ...

where the additional options not covered above are:

`-g` *gPort*

(largely obsolete) *gPort* names a device to use for the display. This was generally used in the past with dual-monitor systems, especially Sun systems in which the layout display might go to `/dev/fb`.

`-i` *tabletPort*

(largely obsolete) *tabletPort* names a device to use for graphics input. This has not been tested with modern graphics tablet devices. It is ignored by the X11 and OpenGL display interfaces.

-F *objFile saveFile*

(largely obsolete) Create an executable file of the current magic process, a core image snapshot taken after all initialization. *objFile* is the name of the original executable, and the image will be saved in *saveFile*. This only works on VAXen and SUNs running an old SunOS (using a.out executables).

Script invocation

Often it is helpful to have a shell script invoke magic with specific options to perform tasks such as generating a GDS file for tapeout. The following example code clip imports GDS into magic as a "vendor cell":

```
magic -dnull -noconsole << EOF
drc off
box 0 0 0 0
load vtop.mag -force
drc off
gds readonly true
gds rescale false
gds read ${cellname}.gds
cellname rename ${cellname} vtmp
load vtmp
select top cell
set pname [lindex [cellname list children] 0]
cellname rename ₩₩₩$pname ${cellname}
select cell ₩₩₩${pname}_0
identify ${cellname}_0
writeall force ${cellname}
quit -noprompt
EOF
```

General window commands (for all windows)

center	closewindow	cursor
help	imacro	logcommands
macro	openwindow	redo
redraw	scroll	setpoint
sleep	specialopen	quit
undo	updatedisplay	version
view	windowborder	windowcaption

windownames	windowscrollbars	xview
zoom	*tk_path_name*	

Layout window commands and window-less commands

addcommandentry	addpath	antennacheck
array	box	calma
caption	cellmanager	cellname
cellsearch	channels	cif
clockwise	closewrapper	contact
copy	corner	crash
crashbackups	crosshair	def
delete	deletecommandentry	display
down	drc	drop
dump	edit	element
erase	expand	ext
ext2sim	ext2spice	extract
extresist	feedback	fill
findbox	findlabel	flatten
flush	garoute	gds
get	getcell	getnode
goto	grid	help
identify	initialize	instance
iroute	irsim	label
lef	load	locking
maketoolbar	measure	move
openwrapper	paint	path
peekbox	plot	plow
polygon	popbox	popstack
port	promptload	promptsave
property	pushbox	pushstack
readspice	render3d	resumeall
rotate	route	save
scalegrid	search	see

select	setlabel	shell
sideways	snap	spliterase
splitpaint	startup	straighten
stretch	suspendall	tag
tech	techmanager	tool *(non-Tcl version)*
tool *(Tcl version)*	unexpand	unmeasure
upsidedown	what	wire
writeall	xload	xor

Netlist window commands

add	cleanup	cull
dnet	dterm	extract
find	flush	join
netlist	pushbutton	print
ripup	savenetlist	shownet
showterms	trace	verify
writeall		

3D window commands

cif	closewindow	cutbox
defaults	help	level
refresh	render	scroll
see	view	zoom

Color window commands

pushbutton	color	load
save		

"Wizard" (developer) layout commands

*bypass	*coord	*extract
*plow	*psearch	*showtech
*tilestats	*tsearch	*watch

"Wizard" (developer) window commands

***crash**	***files**	***grstats**
***pause**	***winddebug**	***winddump**

The Magic Tutorials

The Magic tutorial set has been converted to HTML format, updated, and includes cross-links to commands, tutorial file downloads, and screenshots.

The standard tutorial set:

Tutorial 1: Getting Started
Tutorial 2: Basic Painting and Selection
Tutorial 3: Advanced Painting (Wiring and Plowing)
Tutorial 4: Cell Hierarchies
Tutorial 5: Multiple Windows
Tutorial 6: Design-Rule Checking
Tutorial 7: Netlists and Routing
Tutorial 8: Circuit Extraction
Tutorial 9: Format Conversion for CIF and GDS
Tutorial 10: The Interactive Router
Tutorial 11: Using IRSIM with Magic

(Tutorials kindly translated into Spanish by Ing. Miguel Flores of Don Bosco University can be found here.)
The SCHEME tutorial set:

SCHEME Tutorial 1: The Scheme Command-Line Interpreter
SCHEME Tutorial 2: Boxes and Labels
SCHEME Tutorial 3: Transistor Stacks

The Tcl/Tk tutorial set:

Tcl/Tk Tutorial 1: Introduction
Tcl/Tk Tutorial 4: Simulation with IRSIM

User's Guide Development

To be done:

- Add some general topics, not command-specific.
- Incorporate a lot of the currently text-only material into HTML format.
- More information on the routers and netlists

- Subject index.

email: `tim@bach.ece.jhu.edu`

Last updated: March 30, 2022 at 5:35pm