

ModelSim 5.2 Quick Guide

Web: <http://www.model.com>

Email: support@model.com

sales@model.com

Phone: 503-641-1340



ModelSim Editions

ModelSim-EE Elite Edition (ASIC signoff; Performance; etc.)
ModelSim-PE Personal Edition (FPGA; Basic Functionality)

ModelSim Products

ModelSim-VHDL VHDL Compiler, Simulator, Debugger
ModelSim-VLOG Verilog Compiler, Simulator, Debugger
ModelSim-PLUS VHDL & Verilog Compiler, Simulator, Debugger
ModelSim-LNL VHDL or Verilog Compiler, Simulator, Debugger

ModelSim Definitions

LNL Language Neutral Licensing (VHDL or Verilog)
SKS Single Kernel Simulator

Quick Guide Notes

Find this document at <http://www.model.com/pdf/quickguide.pdf>
 Commands in **bold** are typed at the ModelSim> or VSIM> prompts.

Installation / Environment / Licensing

Documentation

ModelSim Start Here: `<install_dir>/modeltech/docs/ee_start_here.pdf`
 Globetrotter FlexLM Doc: <http://www.globetrotter.com/manual.htm>

Web – Download the Latest Release

<http://www.model.com/support/download.html>
<ftp://ftp.model.com/pub/> (206.103.63.81) [Fast]
<ftp://support.model.com/pub/model.com/> (206.103.57.2) [Slow]

Environment Variables (see ModelSim cmd “printenv”)

LM_LICENSE_FILE	Required	Pathname of <i>license.dat</i> file.
DOPATH	Optional	Search path for “.do” files
EDITOR	Optional	Specifies editor for “edit” cmd.
MODELSIM	Optional	Pathname of <i>modelsim.ini</i> file.
MODELSIM_TCL	Optional	List of <i>modelsim.tcl</i> files.
MODEL_TECH_TCL	Optional	Pathname to Tcl/Tk libraries.
MODEL_TECH	Don't Set	Used internally by ModelSim.
MGC_LOCATION_MAP	Optional	Used as “soft” path to find files.
MTI_TF_LIMIT	Optional	Limits Transcript file size (k).
PLIOBJS	Optional	Used to load PLI object files.
TMPDIR	Optional	Used by VSIM for temp space.

Mentor Graphics Licensing Environment Variable

MGLS_LICENSE_FILE Pathname for Mentor Licensing.

PATH Environment Variable

Unix: Add `<install_dir>/modeltech/bin` to \$path
 PC: PATH will be updated automatically during install.

Starting the License Server

Unix: Copy *license.dat* file to `<install_dir>/modeltech/<platform>/`
 Run `<install_dir>/modeltech/<platform>/START_SERVER`
 PC: Run `<install_dir>/modeltech/win32/flexlm.cpl`
 Use “Setup” and “Control” tabs to configure and start server.

Licensing Diagnostics

Unix: Run `<install_dir>/modeltech/<platform>/lmstat -a & lmdiag`
 PC: Run `<install_dir>/modeltech/win32/lmutil lmstat -a & lmdiag`

Invoking ModelSim

Unix: Run `<install_dir>/modeltech/bin/vsim`
 PC: Start->Programs->Model Tech->ModelSim -or-
 PC: Double-click on: `<install_dir>/modeltech/win32/modelsim.exe`

Wave Window

<left mouse button>	Select signal / Place cursor
<middle mouse button>	Zoom In
<right mouse button>	Zoom Popup Menu
<ctrl-f>	Find next item
<tab> -or- right	Search forward for next edge
<shift-tab> -or- left	Search backward for next edge
i or + o O or -	Zoom in Zoom out
f or F l or L	Zoom full Zoom Last
add wave <item> <item>	Wave specific signals/nets
add wave *	Wave signals/nets in scope
add wave -r /*	Wave all signals/nets in design
add wave -label <name> <item>	Wave and rename a signal/net
add wave abus(31:15)	Wave a slice of a bus
view wave	Display wave window
view wave -new	Display additional wave window
.wave.tree zoomfull	Zoom full
.wave.tree zoomrange <f1> <f2>	Zoom Range
write wave	Print wave window to file

Key ModelSim Commands (see Ref Manual for more)

Command	Where used: (Sh)ell, (M)odelSim> (V)SIM>	Description <i>For details on these commands refer to the ModelSim Reference Manual.</i>
vcom	Sh, M, V	VHDL Compiler (see below)
vdel	Sh, M, V	Deletes a design unit from a specific library
vdir	Sh, M, V	Lists the contents of a library
vlib	Sh, M, V	Creates a design library
vlog	Sh, M, V	Verilog Compiler (see below)
vmap	Sh, M, V	Defines or displays library mappings
vsim	Sh, M, V	VHDL and/or Verilog Simulator (see below)
add button	M, V	Adds a button (i.e. add button MyRun (run 5000))
add list wave	V	Add signals to the List or Wave windows.
add log	V	Log signals to <i>vsim.wav</i> file for analysis later.
alias	M, V	Create a user defined alias (i.e. alias h “history”)
bp, bd	V	Set/Clear a breakpoint (see Managing Breakpoints below).
cd	Sh, M, V	Change directory
change	V	Modify a VHDL variable or Verilog register.
checkpoint	V	Save the state of you simulation (see restore).
configure	M, V	Configure List or Wave window attributes
delete	V	Remove HDL item from List or Wave window
do	M, V	Execute a file of commands (i.e. do macro.do)
drivers	V	Display current and future value of signal or net drivers.
echo	M, V	Display message (i.e. echo “Time is \$now ns.”)
edit	M, V	Invoke editor specified by the EDITOR env variable.
examine	M, V	Examine one or more HDL items (i.e. exa /top/clock)
find	V	Display pathnames of matching HDL items
force	V	Force signals or nets (i.e. force clk 1 10, 0 20 -r 100)
history	M, V	List previous commands
log	M, V	Same as add log above.
.main clear	M, V	Clears the Main window transcript
notepad	M, V	Simple text editor.
printenv	M, V	Display names and values of environment variables.
property	V	Change List or Wave signal attributes (color, radix, etc.)
pwd	M, V	Display current path in Main transcript window.
radix	M, V	Changes the default radix in all windows.
restart	V	Restart the simulator.
restore	M, V	Restore the simulation state from a previous checkpoint .
resume	M, V	Resume macro execution after a pause command.
right left	V	Search in wave window for next transition or -expr.
run	V	Advance simulation time (i.e. run 1000)
search next	V	Search specified window for next item matching pattern.
seetime	V	Scroll List or Wave window to time (ie seetime wave 500)
view	M, V	Open a ModelSim window and pop it to the top.
vsource	V	Display HDL source file in Source window.
when	M, V	Perform action on condition (i.e. when clk=1 {echo clk})
where	M, V	Display info about the environment.
write	M, V	Records names, window contents and preferences to a file.
↑ ↓	M, V	Toggle thru last commands
<ctrl-a>	M, V	Move to beginning of line
<ctrl-e>	M, V	Move to end of line
<ctrl-c>	M, V	Copy the selection in the Main transcript window.
<ctrl-v>	M, V	Paste to the Main transcript window (see <ctrl-c>).
!! !n	M, V	Repeat last command, Repeat nth command
!abc	M, V	Repeat cmd starting “abc”
dumplog64	Sh	Dump the contents of the vsim.wav file in a readable form.
vgencomp	Sh	Create VHDL component from compiled Verilog module.
vmake	Sh	Prints a makefile for a library

vsim

Key Arguments (use –help for full list)

- [-help] Display vsim syntax help
- [-c] Run in cmd line mode
- [-do "cmd" | <file>] Run cmd or file at startup
- [-f <filename>] Pass in args from file
- [-g|G<name=value>] Set VHDL Generic values
- [-hazards] Enable hazard checking
- [-l <logfile>] Save transcript to log file
- [+notimingchecks] Disable timing checks
- [-quiet] Disable loading messages
- [-restore <filename>] Restore a simulation
- [-sdf{ min|typ|max}] Apply SDF timing data i.e.
- <region>=<sdf file> -sdfmin /top=MySDF.txt
- [-sdfnowarn] Disable SDF warnings
- [-t [<mult>]<unit>] Specify time resolution
- [-view <filename>] Log file for VSIM to view
- [-wav <filename>] VSIM log file to create
- [<libname>.<config>] Configuration, Module or
- | <module> Entity/Arch to simulate.
- | <entity>(<arch>)]

Examples

```
vsim top
vsim -lib mywork top -do commands.do
```

vcom

Key Arguments (use –help for full list)

- [-help] Display vcom syntax help
- [-93] [-87] Choose VHDL-1993 or 1987
- [-check_synthesis] Turn on synthesis checker
- [-debugVA] Print VITAL opt status
- [-explicit] Resolve ambiguous overloads
- [-f <filename>] Pass in arguments from file
- [-nocheck] Disable run time range checks
- [-nodebug] Strip internal names
- [-novitalcheck] Disable VITAL95 checking
- [-nowarn <#>] Disable individual warning msg
- [-O0] Disable optimization
- [-quiet] Disable loading messages
- [-refresh] Regenerate library image
- [-work <libname>] Specify work library
- <filename(s)> VHDL file(s) to be compiled

Examples

```
vcom MyDesign.vhd
vcom -93 -work /lib/mylib util.vhd
vcom -refresh
```

vlog

Key Arguments (use –help for full list)

- [-help] Display vlog syntax help
- [-compat] Disable event order optimizations
- [-f <filename>] Pass in arguments from file
- [-hazards] Enable run-time hazard checking
- [-nodebug] Hide internal variables & structure
- [-quiet] Disable loading messages
- [-R <simargs>] Invoke VSIM after compile
- [-refresh] Regenerate lib to current version.
- [-work <libname>] Specify work library
- [-v <library_file>] Specify Verilog source library
- <filename(s)> Verilog file(s) to be compiled

Examples

```
vlog top.v
vlog -work mylib -refresh
```

Managing Breakpoints

- bp** Sets a breakpoint. Without arg shows all **bp**
- bd** Deletes a breakpoint.
- disablebp** Turn off all breakpoints.
- onbreak** Define what to do when a breakpoint is hit during a macro (i.e. **onbreak {resume}**)
- when** Perform actions under certain conditions

Examples

```
bp alu.vhd 147 {do macro.do} Set breakpoint.
bd alu.vhd 147 Clear breakpoint.
when -label when1 {clk'event and b="01100111"} {
echo "Signal c is [examine -bin c]"
stop } Use "when" to show the current whens.
```

Files

- modelsim.ini** System Initialization or Project file. Stores library locations, simulator resolution, paths, etc.
- modelsim.tcl** Window sizes, positions, colors, etc. User Tcl/Tk code.
- startup.do** Default name of macro executed after design is loaded. See "startup =" line in *modelsim.ini*
- transcript** Default filename that ModelSim transcript window activity is saved to.
- vsim.wav** Default name of simulation log file saved by VSIM.

modelsim.ini

Copy modelsim.ini to current directory

Execute **vmap -c**

Loading order (stops after finding first file)

- \$MODELSIM environment variable.
- Current directory if \$MODELSIM is not set.
- In /<install_dir>/modeltech/<platform> directory.
- In /<install_dir>/modeltech directory.

For Detailed Information see:
ModelSim Ref Man "System Initialization/Project File"

modelsim.tcl

Loading order

Always loads: /<install_dir>/modeltech/tcl/vsim/pref.tcl

Loads the first found from:

- \$MODELSIM_TCL if it exists (";" separated list). (all files in list are loaded)
- Current directory ./modelsim.tcl
- \$HOME/modelsim.tcl

Tcl/Tk

Environment Variable
MODELSIM_TCL

Online Documentation
Help->Tcl Help
Help->Tcl Syntax
Help->Tcl Man Pages
Help->Technotes->MTI_Widgets

Language Syntax
command arg1 arg2 arg3 ...

Language Syntax: Commands

```
set <var> <value>
expr <math expression>
exec <ShellCommand>
info <option> <procedure name>
winfo <option> <window name>
```

Language Syntax: Procedures
proc name {arglist} {body}

```
proc diag {a b} {
set c [expr sqrt($a*$a + $b*$b)]
return $c
}
```

Language Syntax: Conditionals
if {boolean} {bodytrue} else {bodyfalse}

```
if { $now < 10000 } {echo $now}
```

Language Syntax: Loops
while {boolean} {body}

```
foreach looVar {valuelist} {cmdBody}
for {initial} {test} {final} {body}
```

Poking around in ModelSim Tcl/Tk

```
info Get info on a Tcl construct.
info xx Find out the args to info.
winfo Get info on Tk widgets.
winfo xx Find out args to winfo.
winfo children . Return the sub-widgets to ModelSim.
lecho [configure wave] Get wave props.
```

Examples

#Print the string length of "Hello, World!"

```
set len [string length "Hello, World!"]
echo "Hello, World! is $len characters long!"
```

#Create a button in the wave window that does something.

```
apply_button_adder wave mBar right red white SayHi {echo hi}
```

#Display the Tcl/Tk source code to apply_button_adder

```
info body apply_button_adder
```

#Set the right mouse button to execute "drivers" on selected signal

```
bind .signals.tree <Button-3> {
set signalnum [.signals.tree index anchor]
set signalline [.signals.tree get2 $signalnum]
set signalname [lindex $signalline 0]
echo [drivers $signalname]
}
```

#Create a separate window containing most used functions:

```
toplevel .hot
frame .hot.run
frame .hot.zoom
pack .hot.run .hot.zoom -side top
button .hot.run.b1 -text "Run 10" -command {run 10}
button .hot.run.b2 -text "Run 100" -command {run 100}
button .hot.run.b3 -text "Run 1000" -command {run 1000}
pack .hot.run.b1 .hot.run.b2 .hot.run.b3 -side left
label .hot.zoom.l1 -text "Zoom: "
pack .hot.zoom.l1 -side left
button .hot.zoom.b1 -text "Full" -command {.wave.tree zoomfull}
button .hot.zoom.b2 -text "4x" -command {WaveZoom .wave out 4.0}
button .hot.zoom.b3 -text "1/4x" -command {WaveZoom .wave in 4.0}
pack .hot.zoom.b1 .hot.zoom.b2 .hot.zoom.b3 -side left
```

#Figure out how to change one of the Run buttons in .hot

```
winfo children .hot
winfo children .hot.run
.hot.run.b2 configure -fg red
.hot.run.b2 configure -text "Run 67"
.hot.run.b2 configure -command {run 67}
```

Standards Supported

VHDL
IEEE 1076-1987
IEEE 1076-1993
VITAL 2.2b & '95

Verilog
IEEE Std 1364-1995
PLI 1.0

Timing
SDF 1.0, 2.0 & 2.1

Value Change Dump
VCD for Verilog and VHDL (EE only)

More Info . . .

Paper and Online (see docs sub-directory)

Start Here [ee_start_here.pdf](#)
Reference Manual [ee_manual_<ver>.pdf](#)
ModelSim Tutorial [ee_tutorial_<ver>.pdf](#)

Technical Notes
www.model.com/support/technote/index.html
See <install_dir>/modeltech/docs/technotes

Company Periodical
ModelUser (req via modeluser@model.com)

ModelSim Help Pulldown
Help->Release Notes
Help->Tcl Man Pages

Training
<http://www.model.com/support/training.html>

This Quick Guide
<http://www.model.com/pdf/quickguide.pdf>

Copyright © 1998,
Model Technology Incorporated
Reproduction Permitted (and encouraged!)
1020-798