# Yosys Installation Guide

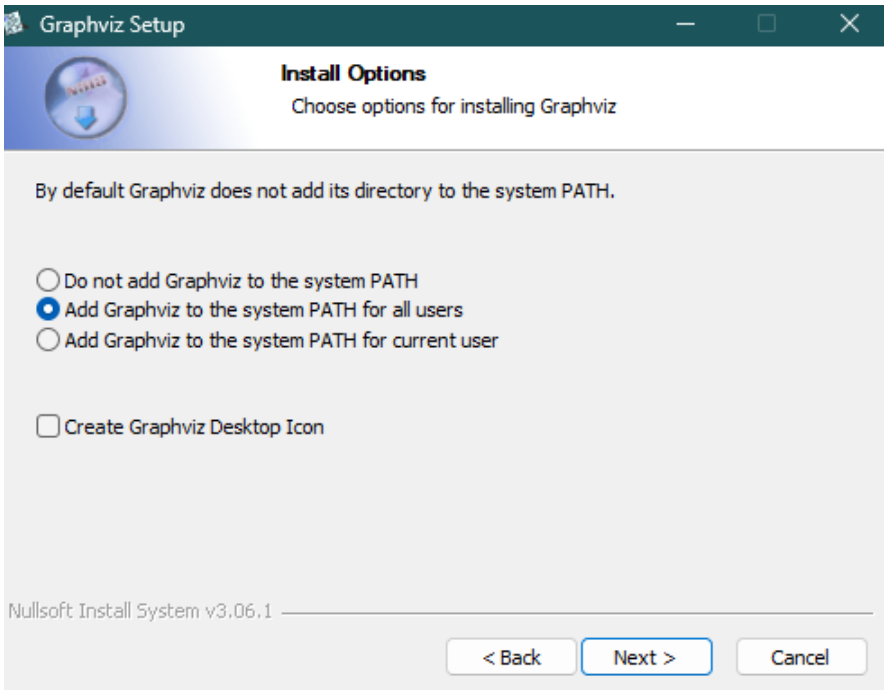
October 1, 2025

## Windows

- **Step 1:** Download Graphviz from this link: Graphviz

  https://gitlab.com/api/v4/projects/4207231/packages/generic/graphviz-releases/14.0.0/windows_10_cmake_Release_graphviz-install-14.0.0-win64.exe

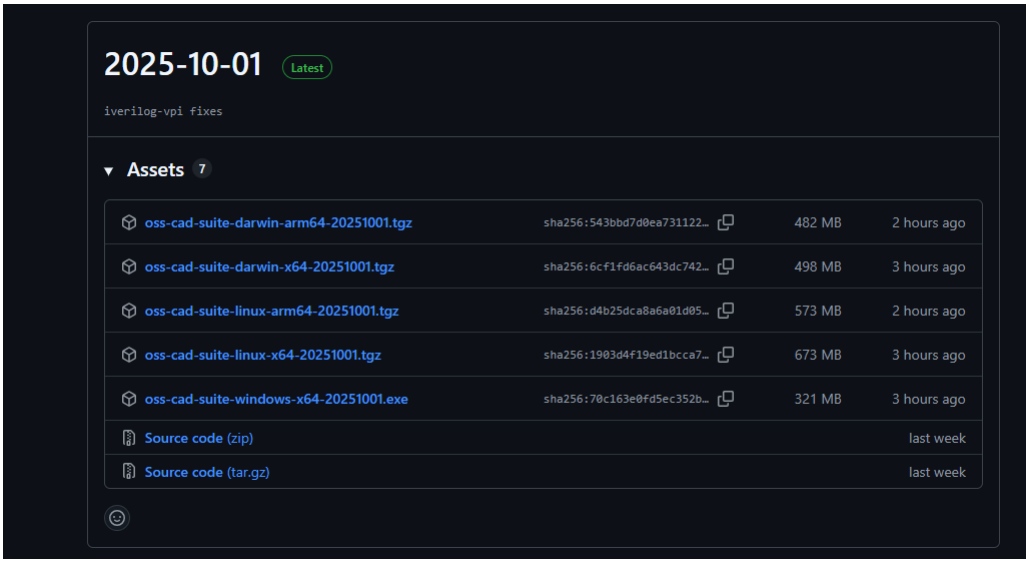
- **Step 2:** Install Graphviz




DON'T FORGET TO ADD TO PATH
Check installation using "dot -v" in terminal, if version is shown install is verified.

- **Step 3:** Download executable from this Yosys HQ

  https://github.com/YosysHQ/oss-cad-suite-build/releases/download/2025-10-01/oss-cad-suite-windows-x64-20251001.exe




Follow the Executable to install Yosys

Follow Running Yosys

# Install the required setup files with sudo.

sudo apt install gcc clang python3 python3-setuptools make git

# Installation

## Linux (Build from Source)

For the latest version:

```
# Install dependencies


sudo apt install build-essential clang bison flex \
   libreadline-dev gawk tcl-dev libffi-dev git \
   graphviz xdot pkg-config python3

# Clone and build
```

(Select File Directory as Desktop)

Step 1: $ git clone --recurse-submodules https://github.com/YosysHQ/yosys.git

Step 2: **cd** yosys

Step 3: make -j$(nproc)

Step 4: sudo make install

Step 5: make **test**

// **test** to confirm successful installation

### Windows (via WSL)

- Install **Windows Subsystem for Linux (WSL)**.
- Choose Ubuntu as your distribution.
- Follow the Linux installation instructions above inside WSL.

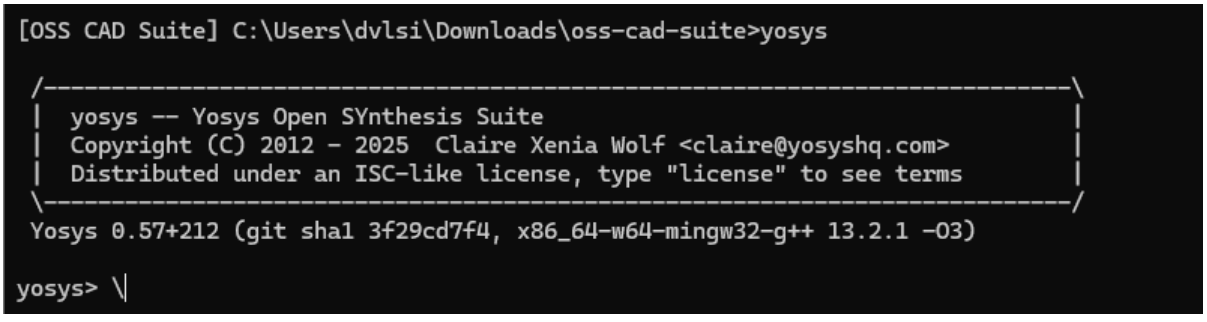
## Running Yosys

### Launching Yosys

Start Yosys with:

./yosys or yosys

This opens the interactive Yosys shell.



## Synthesizing a Verilog File

Assume we have a file `counter.v` or create one syntesizable file:

```
module counter(input clk, input rst, output reg [3:0] out);
  always @(posedge clk or posedge rst) begin
    if (rst)
      out <= 0;
    else
      out <= out + 1;
  end
endmodule
```

To synthesize:

Step 1: initilize yosys

Step 2: **read** –sv <path–to–file>/<file–name>.v

Step 3: hierarchy –top <top–module–name>

(**set** hierarchy to top module without .v extension)

Step 4: write_rtlil

Step 5: proc; opt

Step 6: show \<module–name>

(\<module–name> is required when there are multiple instances)

## Direct Synthesis

YOU NEED TECHNOLOGY FILE FOR THIS

```
yosys –p "read_verilog <full–path>/counter.v; synth; show"
```

## Generating a Netlist

YOU NEED TECHNOLOGY FILE FOR THIS

To write a synthesized netlist:

```
yosys –p "read_verilog <full–path>/counter.v; synth;
   write_verilog counter_netlist.v"
```

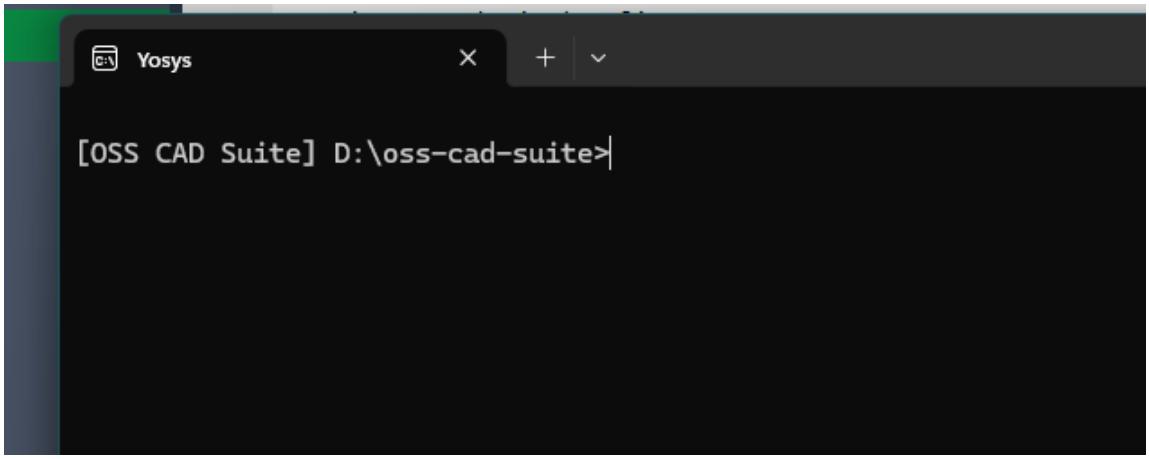
### Printing Synthesis Report

```
stat                                #print synthesis report in terminal

tee -o <file-name>.rpt stat

#create text file for synthesis report
```

### Visualizing RTL Netlist

Open Yosys Directory in Terminal not Yosys suite

```
dot -Tpdf show.dot -o <output-file-name>.pdf
```




This will create the PDF in Yosys Directory

## Flag Explanation

Explanation:

- `read_verilog` loads the Verilog source.
- `write_rtlil` write immediate rtl
- `proc` process translation
- `opt` simple optimization
- `synth` runs generic synthesis.
- `show` opens a schematic viewer.
- `-sv` This is the crucial option. By default, Yosys assumes the input is standard Verilog-2005. Adding -sv tells the Verilog frontend to parse the file using the SystemVerilog syntax and features

- `stat` to print the syntesis stat
- `tee` print the synthesis report in text format
- `ltp` longest topological path