

PyTDL User's Guide

SRC Deliverables Draft, Fall 2009

1. Setting Up The Tools.....	2
1.1. System Requirements.....	2
1.2. Environment.....	2
2. Compiling Designs.....	3
2.1. Basic Compilation.....	3
2.2. Generating Diagrams.....	3

1. Setting Up The Tools

1.1. System Requirements

- Python 2.5.x
- PLY 2.5 or later (given that it is 2.5+ compatible)

PyTDL runs on Python 2.5 ***and is not forward compatible***. The precise revision used in developing this tool is 2.5.2; however, we have no reason to believe that any of the 2.5.x versions will not work. Be warned that we have not tested it in 2.6 or later. Critical introspection components used to compile designs are purported to have changed according to the current documentation (to a different and newer API) or have been deprecated.

Currently, the constraints files are compiled using the **PLY** package, available at: <http://www.dabeaz.com/ply/>. Installation instructions are deferred to the documentation on the source site.

1.2. Environment

If you plan on using PyTDL regularly, it is often quicker to set up a local shell alias. For instance, if you run Bash shell, add the following to your `.bashrc`:

```
alias pytdl="python $LOCATION_TO_PYTDL/PyTDL.py"
```

This makes executing commands easy by simply running “pytdl.”

2. Compiling Designs

2.1. Basic Compilation

Compiling designs is straightforward. Run the following command:

```
python PyTDL.py -o constraintsFile MainDesign.py
```

This will create a sub-directory `build` containing all of the Verilog files generated. From here, you can construct a test bench for simulation. The included tutorials contain sample Verilog test benches which will provide a good starting point.

2.2. Generating Diagrams

PyTDL has unsupported built-in features for generating DOT files used by IBM's GraphViz tool(s). Specify the `-d` flag followed by an output filename to enable this. See the GraphViz documentation on how to use the `dot` tool to then render this file into e.g. PostScript or PDF.