

Build slave documentation

Author: Steven Mohr
Date: \$Date\$
Revision: \$Rev\$
Status: Draft

Contents

[Structure of a build slave directory](#)

[How-To create an Unix slave](#)

[Example for Debian i686](#)

[Ubuntu](#)

[Solaris](#)

[Solaris Py23](#)

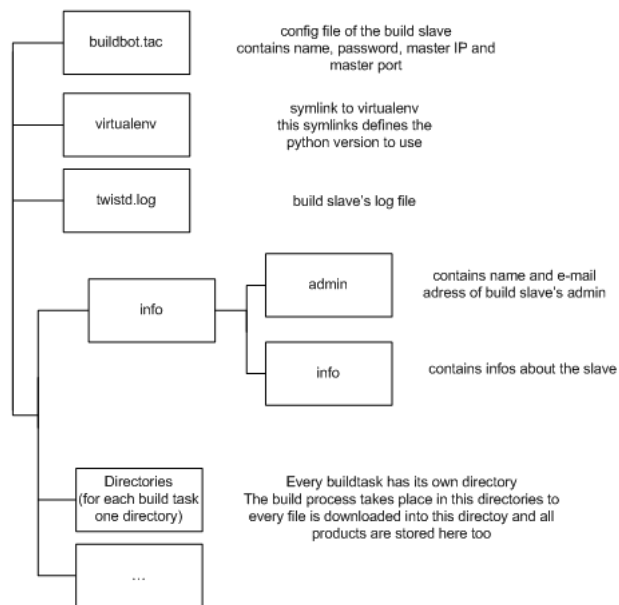
[How-To create a windows slave](#)

[Known problems](#)

Structure of a build slave directory

This is the structure of the directory:

Structure of the build slave directory



How-To create an Unix slave

Example for Debian i686

I've used a netinst image as the base of my builds slave but you can use every image. For the next steps you need to be root:

1. open the `/etc/apt/sources.list` and remove your cdrom drive (else apt-get asks you everytime to insert the install cd).
2. install the following packages via apt-get install

- `zlib1g-dev`
- `build-essential`
- `python-all-dev`
- `python-sqlite2`
- `python-twisted`
- `python-turbogears`
- `subversion`
- `mercurial`
- `libbz2-dev`
- `bzip2`

3. install virtualenv using `easy_install-2.4 virtualenv`

4. rename /usr/bin/virtualenv to /usr/bin/virtualenv-2.4 (the standard python version of my image is 2.4 so I rename it to virtualenv-2.4)
5. install python2.5 and python2.5-dev via apt-get
6. download setuptools install script via wget http://peak.telecommunity.com/dist/easy_install.py
7. run python2.5 easy_install.py
8. easy_install-2.5 virtualenv
9. rename the new virtualenv script (mv /usr/bin/virtualenv /usr/bin/virtualenv-2.5)

I recommend to install buildbot into an own virtual environment. This has a lot of advantages but the biggest is that you can switch back to your normal user so for the next steps you don't need to be root

10. virtualenv-2.5 buildbot
11. activate our new virtual environment: source buildbot/bin/activate
12. easy_install buildbot
13. easy_install twisted

Now we are able to create our first build slave

14. buildbot create-slave slave25 <master> <name> <passwd>

<master> is IPOfTheBuildMaster:Port. <name> is the slave and should fit this pattern: <OS+architecture>_py<python version>, f.e. Ubuntu64_py25. <passwd> is the password to access the build slaves.

15. change directory to slave25 and create a symlink to virtualenv (cd slave25 ln -s /usr/bin/virtualenv-2.5 virtualenv)

We almost have our first builds slave there's only one thing we have to do before we're finished: Because of an limitation in the mercurial class of BuildBot we have to create 2 directories.

16. mkdir TG2full, mkdir TG2full/build, mkdir TG2full/build/webhelpers TG2full/build/pylons

Our first slave is finished. The second is as easy as the first, so let's start:

17. buildbot create-slave slave24 <master> <name> <passwd>

<master> is the same as before. You only have to change the version number in <name> and define a new password.

18. change directory to slave24 and create a symlink to virtualenv (cd slave24 ln -s /usr/bin/virtualenv-2.4 virtualenv)
19. mkdir TG2full, mkdir TG2full/build, mkdir TG2full/build/webhelpers TG2full/build/pylons
20. switch back to the root directory of our slaves

The creation of our last slave isn't that easy. The first problem is that there aren't any Py2.3 packages in the repo, so you have to build it from source.

21. `wget http://www.python.org/ftp/python/2.3.7/Python-2.3.7.tgz`

22. `untar -xf Python-2.3.7.tgz`

23. `cd Python-2.3.7`

24. `./configure`

For the next steps you need to be root:

25. `make`

26. `make install`

27. `python23 ez_setup.py`

28. `easy_install-2.3 virtualenv==0.9`

29. rename the new virtualenv script (`mv /usr/local/bin/virtualenv /usr/local/bin/virtualenv-2.3`)

30. remove `/usr/local/bin` from `$PATH`

31. `buildbot create-slave slave23 ...`

32. change directory to `slave23` and create a symlink to `virtualenv` (`cd slave23 ln -s /usr/local/bin/virtualenv-2.3 virtualenv`)

So that's almost everything; one last step

33. download `tg_sbuildsteps.py` from svn (http://svn.turbogears.org/build/buildmaster/tg_sbuildsteps.py) and copy it to (directory of your python installation)/lib/python25/site-packages/buildbot-0.7.7-py2.5.egg/buildbot/slaves. Rename `commands.py` to `_commands.py` and run `cat _commands.py tg_mbuildsteps.py > commands.py`

That's it, you have your first three build slave. Now you can start them with `buildbot start <DirOfSlave>`.

Ubuntu

Under Ubuntu it's the same the only difference is that you have to use `sudo` for all root commands.

Solaris

I've used OpenSolaris to create the image. It's important to use the Developer Edition. This includes a few packages which you really need.

The first step is to download and install all packages you need for Python, Subversion and Mercurial. It's:

- `apache2`

- bzip2
- db-4.2
- expat
- gcc-3.4.6
- gdbm
- libiconv
- libintl
- libxml
- mercurial
- ncurses
- neon
- openssl
- python-2.5.1
- readline
- sqlite
- subversion
- swig
- tcl
- tk
- wget
- zlib

This packages are at sunfreeware.com.

The next steps are similar to Debian. You have to install virtualenv, rename it, install setuptools for Py2.5, install virtualenv, rename it, install BuildBot and Twisted, create the slaves, create directories and symlinks, download and copy tg_sbuildsteps.py and start slaves.

1. easy_install-2.4 virtualenv
2. mv /usr/bin/virtualenv /usr/bin/virtualenv-2.4
3. wget http://peak.telecommunity.com/dist/ez_setup.py
4. /usr/local/bin/python2.5 ez_setup.py
5. /usr/local/bin/easy_install-2.5 virtualenv
6. mv /usr/local/bin/virtualenv /usr/local/bin/virtualenv-2.5
7. buildbot create-slave slave25 <master> <name> <passwd>
8. buildbot create-slave slave24 <master> <name> <passwd>

The next three steps have to be done for both slave directories: 9. switch to slave directory 10. create symlink to virtualenv (ln -s /usr/(local/)bin/virtualenv-2.(4 or 5) virtualenv) 11. create directories

- mkdir TG2full
 - mkdir TG2full/build
 - cd TG2full/build
 - mkdir pylons webhelpers
12. download tg_sbuildsteps.py from svn (http://svn.turbogears.org/build/buildmaster/tg_sbuildsteps.py) and copy it to (directory of your python installation)/lib/python25/site-packages/buildbot-0.7.7-py2.5.egg/buildbot/slaves. Rename commands.py to _commands.py and run
cat _commands.py tg_mbuildsteps.py > commands.py

So now you are finished. You have 2 slaves with you can start with buildbot start <DirOf-Slave>. If you're building dependencies there could be a problem. The first time you run a build I've got the error that the rsa key of some SVN repo is unknown and that you have to add it manually. In this case you have to add it manual and start the build again. I will try to list the URLs you have to add but if you get this error you know what to do.

Solaris Py23

I've stopped working on the Py23 slave for Solaris because I was not able to build Python2.3 on this OS. If I have time after the creation of the Windows slave, I create this slave.

How-To create a windows slave

To create a slave with Windows is easy. The how-to from buildbot.net works well. So let's start:

First download some installer:

1. Visit python.org and download the installer for Python 2.3, 2.4 and 2.5.
2. Download pyWin32 extensions from <https://sourceforge.net/projects/pywin32/> and MinGW from MinGW <http://mingw.org>
3. Download Twisted for Py2.4 from <http://twistedmatrix.com/>
4. Download BuildBot as a zipfile
5. Run all installer
6. extract buildbot zip and run the setup.py with the install argument with Py2.4.
Then you have to edit the buildbot.bat and change the python directroy to python24

Edit the environment variables

7. add .py to PATHEXT
8. add the python24 and python24scripts directory to PATH

Install easy_install and virtualenv

9. download eay_install from peak.telecommunity.com/dist/ez_setup.py

10. copy this script in all three python directories
11. open cmd and run this script with the different python versions
12. run easy_install virtualenv with the different python versions

Create the slaves

13. buildbot create-slave slave25 <master> <name> <passwd>
14. buildbot create-slave slave24 <master> <name> <passwd>
15. buildbot create-slave slave23 <master> <name> <passwd>

The creation of the slaves is similar to UNIX OS.

Then we have to create some files. 16. Like the symlinks on a unix-like OS, we need something comparable in Windows.

Unexpected indentation.

We create a batch file with this content and the name virtualenv.bat in every slave directory: C:\python2[3,4,5]\scripts\virtualenv %*

Block quote ends without a blank line; unexpected unindent.

17. download tg_sbuildsteps.py from svn (http://svn.turbogears.org/build/buildmaster/tg_sbuildsteps.py) and append its content to the commands.py in lib/python25/site-packages/buildbot-0.7.7-py2.5.egg/buildbot/slaves of your Python24 directory.

Our last step is to create the directories for the Mercurial checkouts. We need a TG2fullbuildwebhelpers and a TG2fullbuildpylons directory in every slave directory.

Known problems

- Buildbot needs a stable connection. It's not recommended to run a build slave with a WiFi connection. This will spam Buildbot's waterfall display with connect/disconnect messages.