

# **Manual del entorno de integración continua**

Última versión del documento:

<http://oriolrius.cat/blog/2010/04/22/integracio-continua>

Actualizaciones:

2010/04/22 - Oriol Rius - Inicial

Autor:

Oriol Rius <[oriol@joor.net](mailto:oriol@joor.net)> - <http://oriolrius.cat>

Agradecimientos:

Miquel Torres <[tobami@gmail.com](mailto:tobami@gmail.com)>

# Índice

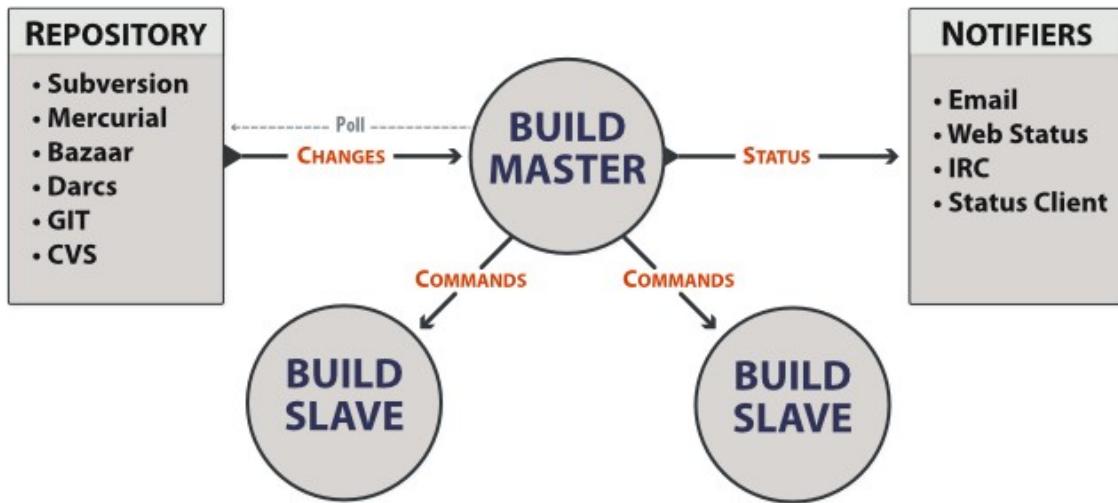
Manual del entorno de integración continua.....	1
El código.....	3
Como funciona buildbot.....	4
Configurar buildbot master.....	4
Consultar estado de los tests.....	5
Menú principal.....	5
Grid display.....	6
Transposed grid display.....	7
Waterfall display.....	8
Lastest build.....	9
Recent builds.....	10
Buildslave.....	11
ChangeSource.....	12
Profiling.....	12
About buildbot.....	12
Consultar evolución del rendimiento.....	13
Menú.....	13
Overview.....	14
Timeline: all in a grid.....	15
Timeline: de un benchmark.....	16
Parte de administración.....	17
Esquema de base de datos.....	18
Añadir datos a la aplicación.....	19
Enlaces relacionados.....	20
Apendice I: Código de demo (buldbot-test).....	21
test/test.py.....	21
test/tests.py.....	22
test/profiling.py.....	22
setup.py.....	23
Apendice II: master.cfg.....	24

## El código

- Debemos programar de forma natural
- Hay que pensar que es vital probar lo que programamos
- Unittest o cualquier otro sistema similar son buenas opciones para validar el comportamiento de nuestro código
- Es vital pensar en que el sistema de test debe ser lo más automático posible
- A veces no es suficiente que las cosas devuelvan el resultado esperado
- Hay que pensar en problemas colaterales como:
  - perdida de rendimiento
  - perdidas de memoria
  - etc
- Así pues nos puede interesar apoyarnos en herramientas como **guppy-pe** y el módulo **resource** de python para controlar el rendimiento de nuestro código

Código de ejemplo: master.cfg (Apendice II)

# Como funciona buildbot



## Configurar buildbot master

El fichero de configuración lo podemos dividir en:

- BUILDSLAVES: declaración de build slaves
- CHANGESOURCES: declaración de repositorios de código y cada cuanto tiempo se deben consultar, también se pueden usar triggers desde el código para notificar cambios.
- SCHEDULERS: recibe los cambios y lanza tareas, en esencia es el que decide a partir de cuantos cambios, tiempo, triggers, etc. debe lanzar un proceso de test.
- BUILDERS: cada builder está formado por la suma de sus buildsteps, que se podría traducir por acción. Estas acciones se pueden asociar a uno o varios build-slaves, que son quienes ejecutarán las acciones. Por ejemplo, una acción puede ser hacer un **checkout** del código, otra acción compilarlo, otra testearlo, etc.
- STATUS TARGETS: hay diversas formas de notificar la actividad del buildbot, la más común es publicar una web donde consultar la evolución de los cambios. Pero también se puede notificar vía email, IRC, etc.
- DEBUGGING OPTIONS: interfaces para depuración interactiva del master y de los buildsteps.
- PROJECT IDENTITY: información referente al proyecto.

# Consultar estado de los tests

Se describen a grandes rasgos los datos disponibles a través de la interface web de buildbot.

## Menú principal



## buildbot

- the [Grid Display](#) will give you a developer-oriented summary of recent builds.
- the [Transposed Grid Display](#) presents the same information as the grid, transposed.
- the [Waterfall Display](#) will give you a time-oriented summary of recent builds.
- The [Latest Build](#) for each builder is here.
- [Recent Builds](#) are summarized here, one per line.
- [Buildslave](#) information
- [ChangeSource](#) information.
- [Profiling](#) application.
- [About this Buildbot](#)

## Grid display

Podemos ver como han ido los últimos tests, a lo largo de las diferentes revisiones.

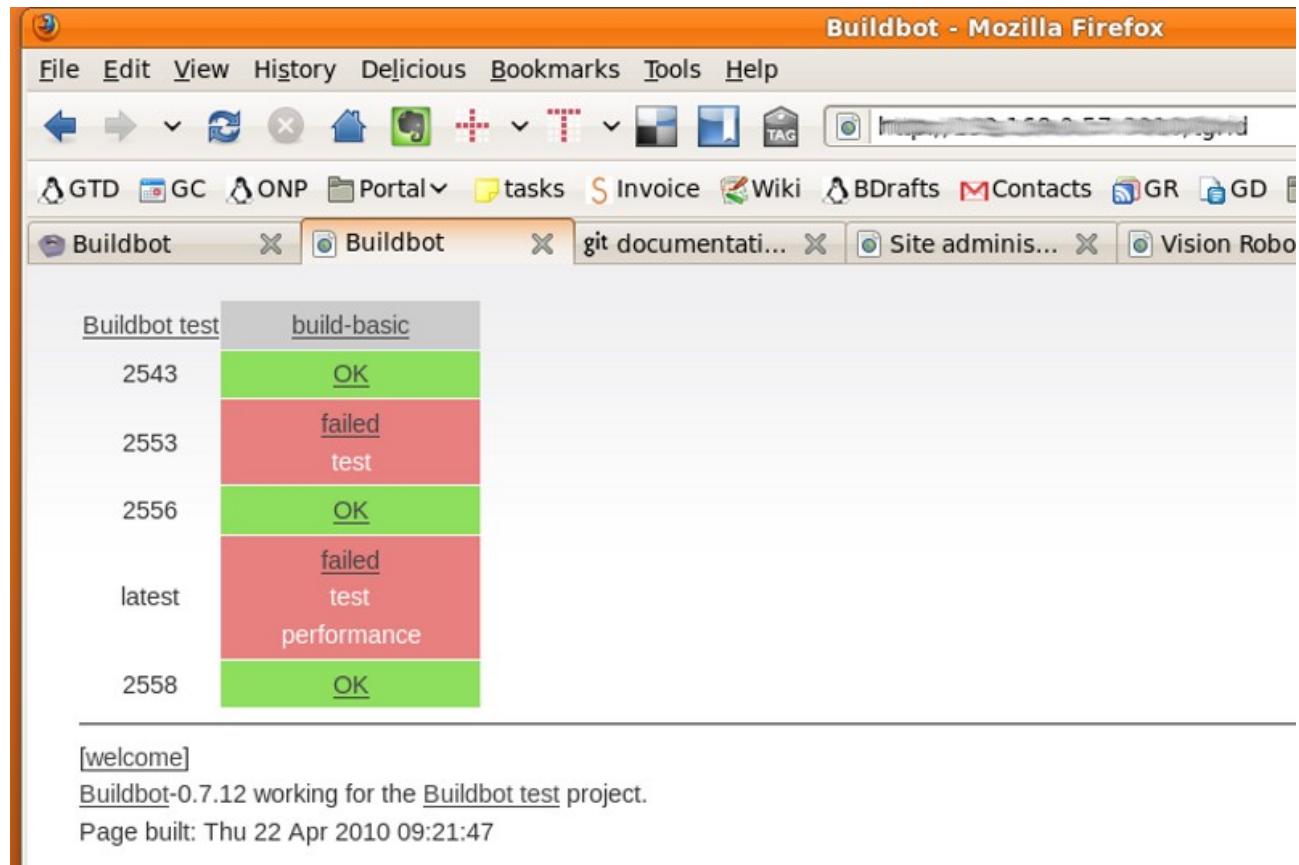
The screenshot shows a Mozilla Firefox browser window with the title "Buildbot - Mozilla Firefox". The address bar displays the URL "http://192.168.0.57:8080/grid". The toolbar includes standard icons for back, forward, search, and refresh. The menu bar has options like File, Edit, View, History, Delicious, Bookmarks, Tools, and Help. The bookmarks bar contains links to GTD, GC, ONP, Portal, tasks, Invoice, Wiki, BDrafts, Contacts, GR, and GD. The main content area shows a grid of build test results for the "Buildbot test" project across six revisions: 2543, 2553, 2556, latest, and 2558. The grid consists of six columns with the following data:

Buildbot test	2543	2553	2556	latest	2558
<a href="#">build-basic</a>	<a href="#">OK</a>	<a href="#">failed test</a>	<a href="#">OK</a>	<a href="#">failed test performance</a>	<a href="#">OK</a>

Below the grid, a message reads: "[welcome] Buildbot-0.7.12 working for the [Buildbot test](#) project. Page built: Thu 22 Apr 2010 09:21:22".

## Transposed grid display

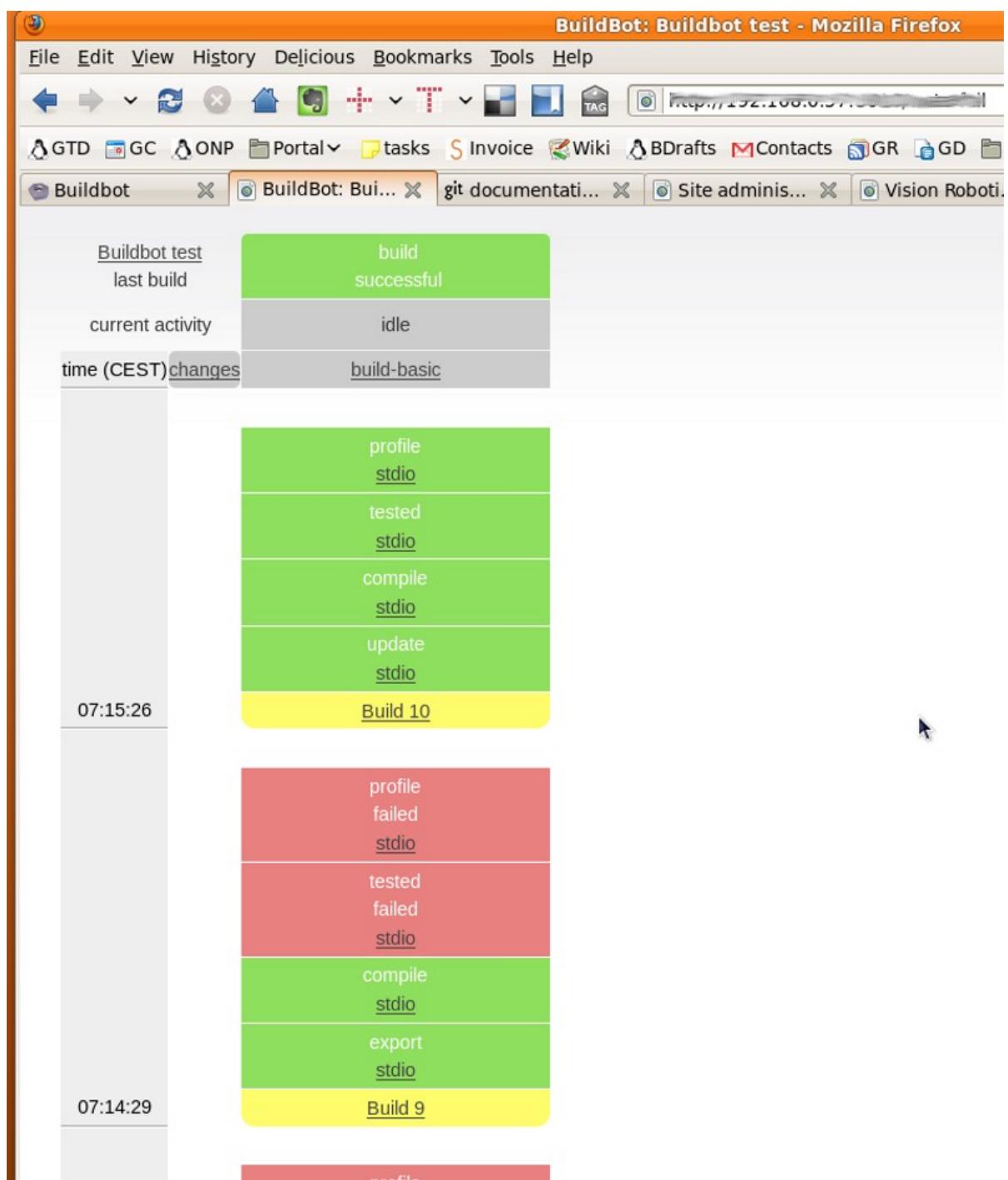
Muestra la misma información que el punto anterior pero las revisiones se organizan en vertical y no en horizontal.



## Waterfall display

Cuadro resum con muchas información:

- resultado de los últimos test organizados por 'build'
- resultados parciales de la revisión, de cada uno de los buildstep
- acceso directo a la salida (status) de cada uno de los buildstep
- referencia horaria de cuando se realizó el 'build'
- estado de la actividad actual



## Lastest build

Podemos consultar el estado del último test y además forzar la ejecución de un nuevo test con los parámetros de entorno que nos interesen.

The screenshot shows a Mozilla Firefox browser window titled "Latest Build - Mozilla Firefox". The address bar contains a URL starting with "http://...". The toolbar has various icons for navigation and bookmarks. The main content area displays a "Latest builds:" section. It shows a green box for "build-basic" with the number "2558", the word "build", and "successful". To its right is a grey box labeled "idle". Below this, there is a form with fields for "Your name:", "Reason for build:", "Branch to build:", "Revision to build:", "Property 1 Name:" (with a value field), "Property 2 Name:" (with a value field), "Property 3 Name:" (with a value field), and a "Force Build" button. A cursor arrow is visible on the right side of the page.

Latest Build - Mozilla Firefox

File Edit View History Delicious Bookmarks Tools Help

Buildbot Latest Build git documentati... Site adminis... Vision Roboti... We

Latest builds:

build-basic 2558 build successful idle

To force a build on all Builders, fill out the following fields and push the 'Force Build' button

Your name:

Reason for build:

Branch to build:

Revision to build:

Property 1 Name:  Value:

Property 2 Name:  Value:

Property 3 Name:  Value:

Force Build

## Recent builds

- Resumen de los resultados de los últimos tests, además de relacionar la hora, la revisión y el número de build.
- También se puede forzar la re-ejecución de alguno de los tests.

Recent Builds - Mozilla Firefox

File Edit View History Delicious Bookmarks Tools Help

Back Forward Stop Home Search Bookmarks Tools Help

Recent Builds

Last 20 finished builds:

- (Apr 22 07:15) rev=[2558] success build-basic #10: build successful
- (Apr 22 07:14) rev=[??] failure build-basic #9: failed test performance
- (Apr 21 20:55) rev=[??] failure build-basic #8: failed test performance
- (Apr 21 20:52) rev=[??] failure build-basic #7: failed test performance
- (Apr 21 17:45) rev=[2556] success build-basic #6: build successful
- (Apr 21 17:38) rev=[2553] failure build-basic #5: failed test
- (Apr 21 10:29) rev=[2543] success build-basic #4: build successful
- (Apr 21 10:23) rev=[2541] failure build-basic #3: failed compile
- (Apr 21 10:18) rev=[2541] failure build-basic #2: failed compile
- (Apr 20 17:48) rev=[2531] success build-basic #1: build successful
- (Apr 20 17:19) rev=[2527] success build-basic #0: build successful

To force a build on all Builders, fill out the following fields and push the 'Force Build' button

Your name:

Reason for build:

Branch to build:

Revision to build:

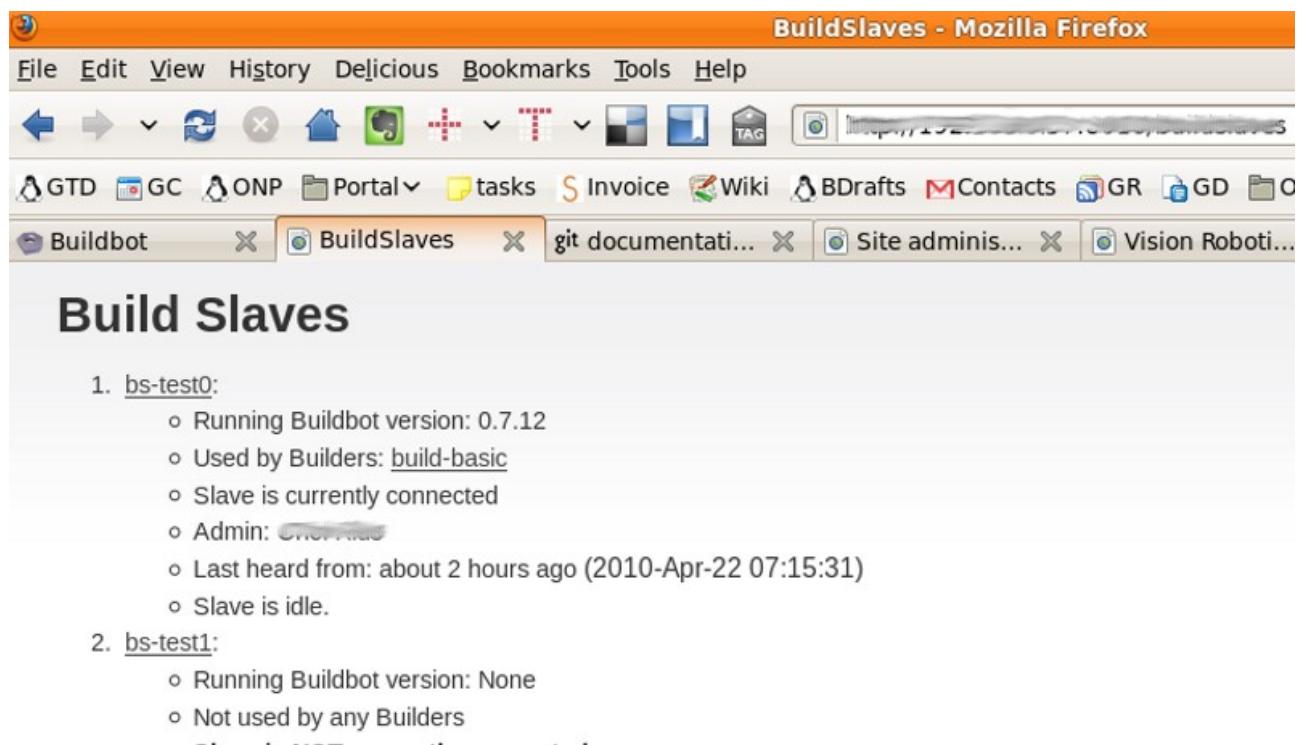
Property 1 Name:  Value:

Property 2 Name:  Value:

Property 3 Name:  Value:

## Buildslave

Descripción y estados de los buildslaves.



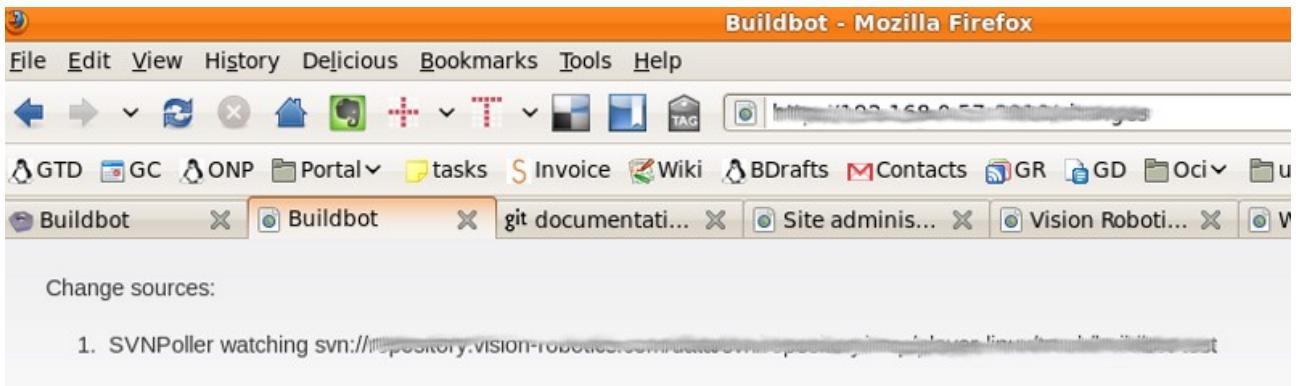
The screenshot shows a Mozilla Firefox browser window titled "BuildSlaves - Mozilla Firefox". The address bar displays a URL starting with "http://127.0.0.1:8000/buildslaves". The toolbar includes standard buttons for back, forward, search, and refresh, along with links to "GTD", "GC", "ONP", "Portal", "tasks", "Invoice", "Wiki", "BDrafts", "Contacts", "GR", "GD", and "O". The tab bar shows several open tabs, with the "BuildSlaves" tab currently active. The main content area displays a list of build slaves:

### Build Slaves

1. bs-test0:
  - o Running Buildbot version: 0.7.12
  - o Used by Builders: build-basic
  - o Slave is currently connected
  - o Admin: clerk
  - o Last heard from: about 2 hours ago (2010-Apr-22 07:15:31)
  - o Slave is idle.
2. bs-test1:
  - o Running Buildbot version: None
  - o Not used by any Builders
  - o **Slave is NOT currently connected**

## ChangeSource

Ruta de los repositorios que estan siendo vigilados.



## Profiling

Es un enlace 'hardcodeado' en el template 'index.html' de la aplicación para poder acceder directamente a la aplicación **codespeed**.

## About buildbot

Información sobre la aplicación.

A screenshot of a Mozilla Firefox browser window titled "About this Buildbot - Mozilla Firefox". The address bar shows a URL starting with "http://122.168.0.57:8080/about". The tab bar has several tabs open, including "Buildbot" (which is active), "About this B...", "git documentati...", "Site adminis...", "Vision Roboti...", and others. The main content area displays the text "Welcome to the Buildbot" and "Version Information". Below that is a bulleted list of system details: "Buildbot: 0.7.12", "Twisted: 10.0.0", "Python: 2.4.3 (#2, Jan 21 2010, 19:56:43) [GCC 4.0.3 (Ubuntu 4.0.3-1ubuntu5)]", and "Buildmaster platform: linux2".

Welcome to the Buildbot

Version Information

- Buildbot: 0.7.12
- Twisted: 10.0.0
- Python: 2.4.3 (#2, Jan 21 2010, 19:56:43) [GCC 4.0.3 (Ubuntu 4.0.3-1ubuntu5)]
- Buildmaster platform: linux2

Source code

Buildbot is a free software project, released under the terms of the [GNU GPL](#).

Please visit the [Buildbot Home Page](#) for more information, including documentation, bug reports, and source downloads.

# Consultar evolución del rendimiento

Esta información esta almacenada en una base de datos MySQL y a través del código de **codespeed** con algunos pequeños cambios hemos desarrollado nuestra propia herramienta de seguimiento del rendimiento.

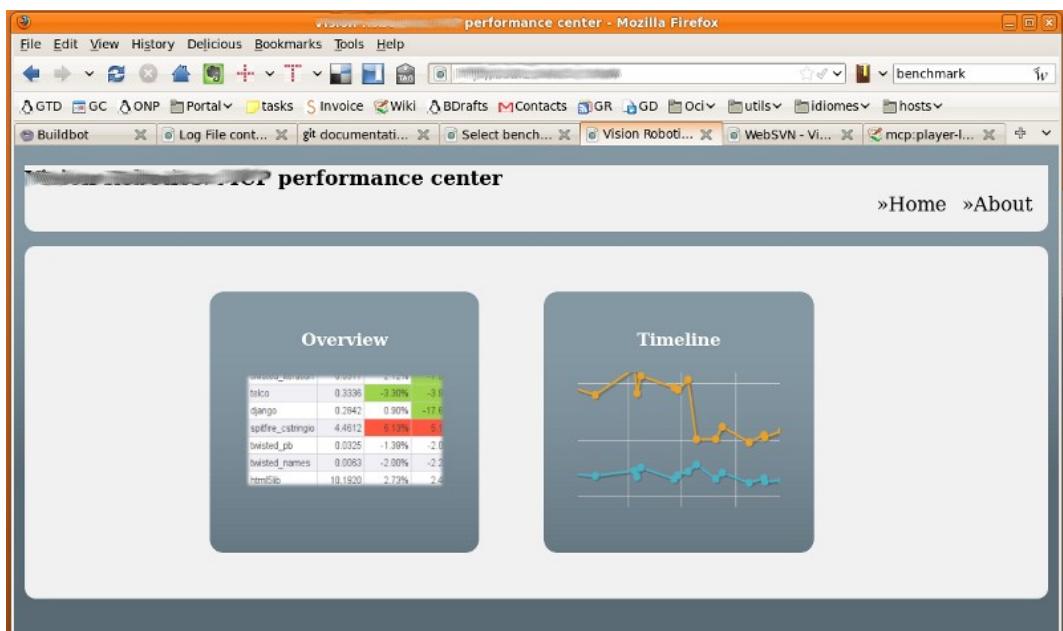
Funcionalidades:

- Para un ejecutable
- Podemos tener diversos tests (benchmark)
- Sobre cada benchmark guardamos un valor asociado a su revisión
- Consultar una tabla con las tendencias de los benchmark para un ejecutable
- Consultar una gráfica con la evolución de los valores de un benchmark a lo largo del tiempo

## Menú

Se debe escoger:

- **overview**: tabla con las tendencias de los benchmark para un ejecutable
- **tiimeline**: gráfica con la evolución de los valores de un benchmark a lo largo del tiempo



## Overview

- Se puede elegir ejecutable a consultar
- Host sobre el que se han hecho las pruebas
- Resultados para una revisión concreta
- Tabla con los benchmark y las tendencias
- Histórico de versiones con sus notas del repositorio

The screenshot shows the MCP performance center interface. The top navigation bar includes links for File, Edit, View, History, Delicious, Bookmarks, Tools, and Help. Below the menu is a toolbar with various icons. The main title is "Vista Preliminar MCP performance center". On the right, there are links to "»Home" and "»About". The central area has tabs for "Overview" (which is selected) and "Timeline".  
  
The "Executable" section shows a radio button for "profiling" selected, while "test" is unselected.  
  
The "Options" section includes a "Trend:" dropdown set to "dotted" and a "Compare to:" dropdown showing a single entry.  
  
The "Host" section shows a radio button for "bs-test0" selected, with "bs-test1 (verybox+broadcom hw decoder)" as an alternative.  
  
The main content area displays results for revision 2010-04-21 20:02:40 - 2558. It features a table with columns: Benchmark, Units, Value, std dev, Current change, Trend, Times, and slower (log2 scale). The table contains three rows:

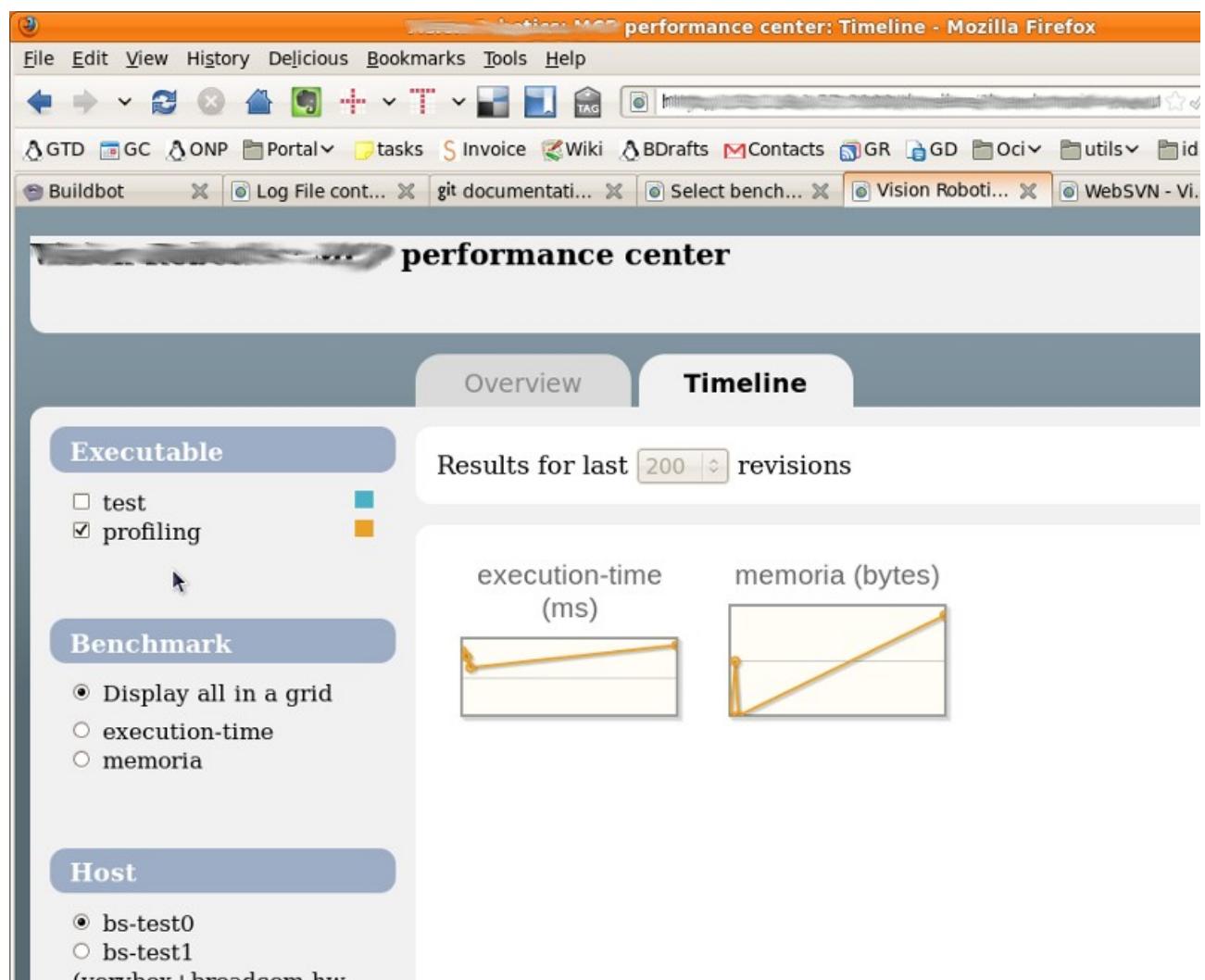
Benchmark	Units	Value	std dev	Current change	Trend	Times	slower (log2 scale)
memoria	bytes	1753536.0000		4383740.00%	2435366.67%		4x
execution-time	ms	0.0600		46.34%	9.09%		
Average				2191893.17%	1217687.88%		

  
Below the table is a section titled "Commit logs for trunk at svn://repository.vision-robotics.com/data/svn/repository/mcp/player-linux/trunk/buildbot-test" which lists three commits by oriolrius:

Date	Message
2010-04-21 20:02:40	oriolrius committed 2558 scheduler
2010-04-21 17:43:49	oriolrius committed 2556 random buffer size
2010-04-21 17:41:41	oriolrius committed 2555 montintoritzem el buffer

## Timeline: all in a grid

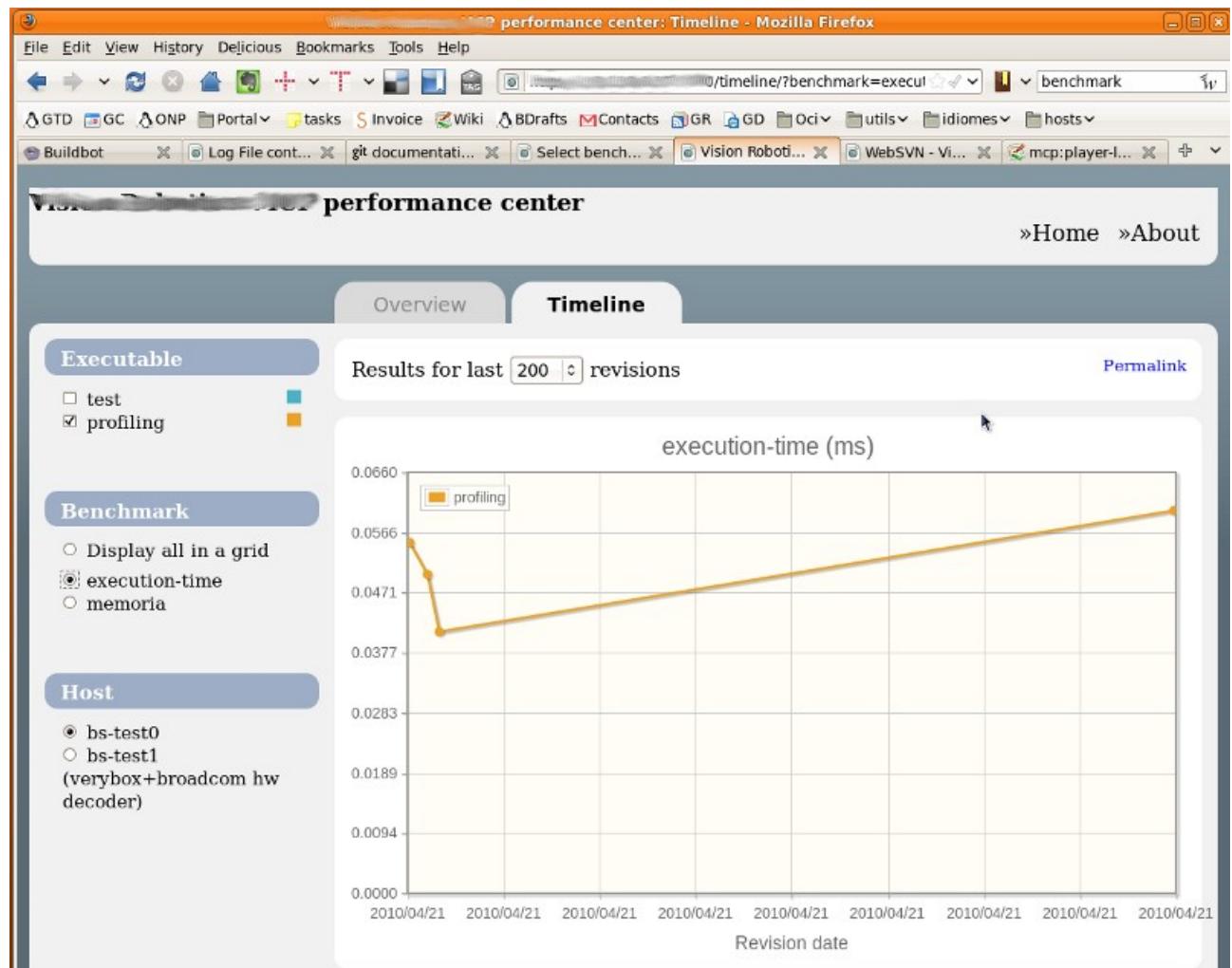
Sobre un ejecutable se muestran en miniatura las gráficas de los diferentes benchmark.



## Timeline: de un benchmark

Se escoje un ejecutable y un benchmark, podemos ver con más detalle la evolución de los valores del mismo.

También se puede filtrar por las 'n' revisiones anteriores y por un 'host'.



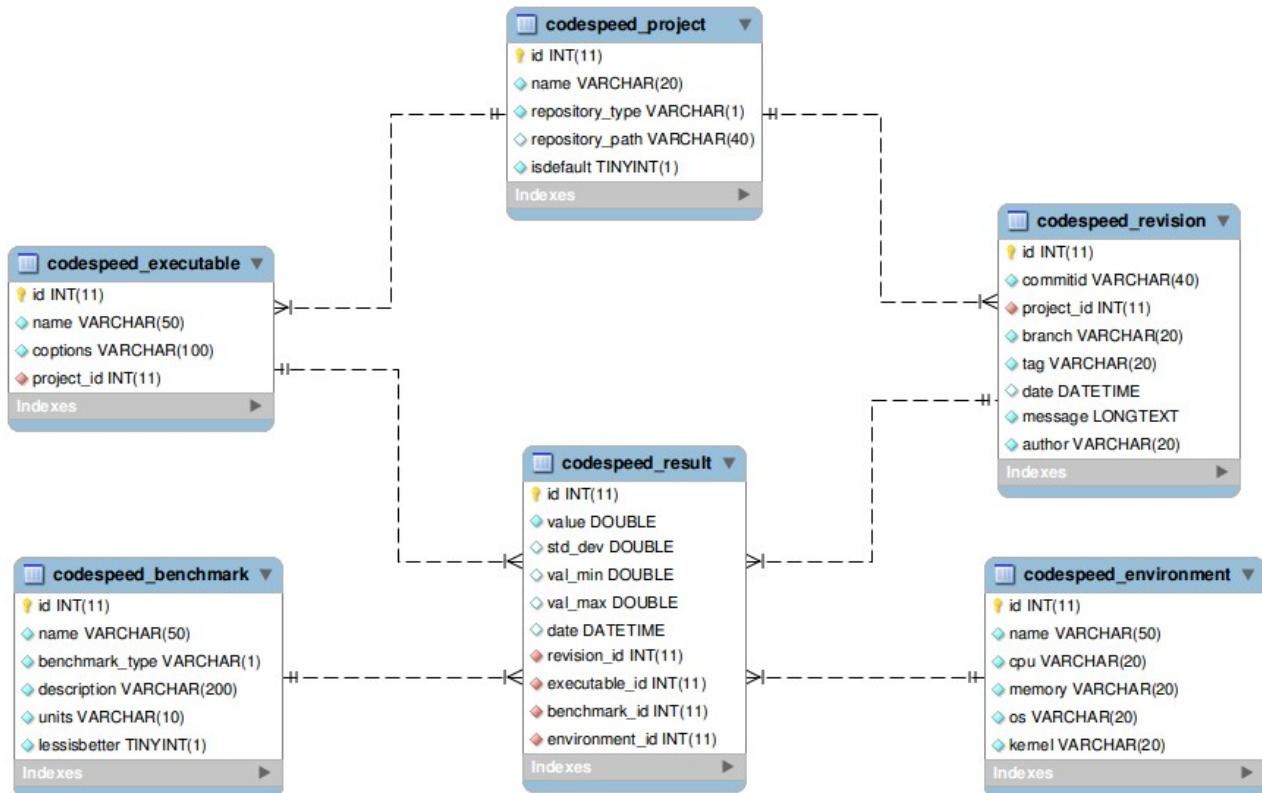
## Parte de administración

Si accedemos a la URL/admin, hay una vista sobre las tablas que usa la aplicación y podemos configurar los parámetros con los que trabajará la aplicación.

The screenshot shows the Django Site administration interface. The browser title bar reads "Site administration | Django site admin - Mozilla Firefox". The address bar shows the URL "http://127.0.0.1:8000/admin/". The toolbar includes standard browser buttons like back, forward, and search. The tab bar has several open tabs, including "Buildbot", "Log File cont...", "git documentati...", "Site adminis...", and "Vision Roboti...". The main content area is titled "Django administration" and contains a "Site administration" heading. On the left, there's a sidebar with three main sections: "Auth" (Groups, Users), "Codespeed" (Benchmarks, Environments, Executables, Projects, Results, Revisions), and "Sites" (Sites). Each item in the sidebar has an "Add" and a "Change" link. To the right of the sidebar is a "Recent Actions" panel listing various actions taken, such as "execution-time Benchmark", "test none Executable", and multiple entries for "Result object Result".

## Esquema de base de datos

Con esta vista podemos tener claro de un vistazo cuales son los datos que se almacenan en la herramienta y sus relaciones.



## Añadir datos a la aplicación

Un snippet de código muy simple para entender como se pueden añadir datos a **codespeed**:

```
#!/usr/bin/python
from datetime import datetime
import urllib, urllib2

SPEEDURL = 'http://IP:8000/' # This will be pyspeed.pypy.org/
data = {
    'commitid': '2539',
    'project': 'buildbot-test',
    'branch': 'trunk',
    'executable_name': 'basic',
    'executable_options': 'none',
    'benchmark': 'memoria',
    'environment': 'bs-test0',
    'result_value': '201',
    'result_date': datetime.today(),
}

def add(data):
    params = urllib.urlencode(data)
    f = None
    response = "None"
    print "revision %s, benchmark %s" % (data['commitid'], data['benchmark'])
    try:
        f = urllib2.urlopen(SPEEDURL + 'result/add/', params)
        response = f.read()
        f.close()
    except urllib2.URLError, e:
        if hasattr(e, 'reason'):
            response = '\n We failed to reach a server\n'
            response += ' Reason: ' + str(e.reason)
        elif hasattr(e, 'code'):
            response = '\n The server couldn\'t fulfill the request\n'
            response += ' Error code: ' + str(e)
    print "Server (%s) response: %s\n" % (SPEEDURL, response)

if __name__ == "__main__":
    add(data)
```

## Enlaces relacionados

- Buildbot docs: <http://djmitche.github.com/buildbot/docs/latest/>
- guppy-pe:
  - <http://guppy-pe.sourceforge.net/>
  - <http://www.apparatusproject.org/blog/tag/heapy/>
- resource.getresusage():
  - <http://docs.python.org/library/resource.html#resource.getrusage>
  - <http://www.doughellmann.com/PyMOTW/resource/>
- speedcode:
  - <http://speed.pypy.org/>
  - <http://wiki.github.com/tobami/codespeed/>
- setuptools: <http://peak.telecommunity.com/DevCenter/setuptools>
- unittest: <http://docs.python.org/library/unittest.html>

# Apendice I: Código de demo (buldbot-test)

## test/test.py

```
#!/usr/bin/python
import random

class Test():
    def demo(self):
        if random.choice([True, False]):
            return "hello world!"
        else:
            return "bye bye"
    # memory leak
    buf=''

    def __init__(self):
        self.buf=''
    def get_buff(self):
        return self.buf;
    def size(self):
        return len(self.buf)
    def add(self):
        self.buf+=self.random_string()
    def random_string(self):
        string=''
        for count in xrange(1,random.randint(10000,100000)):
            string+='0123456789'
        return string

def main():
    var = Test()
    print var.demo()

if __name__ == "__main__":
    main()
```

## test/tests.py

```
#!/usr/bin/python
import unittest
from test import Test

class ProvarTest(unittest.TestCase):
    def runTest(self):
        """ comprobamos que se devuelve un hello world """
        var=Test()
        self.assertEqual(var.demo(),'hello world!', 'no devuelve hello world')

if __name__ == "__main__":
    unittest.main()
```

## test/profiling.py

```
#!/usr/bin/python
import resource
from guppy import hpy
import random
from test import Test
from datetime import datetime
import urllib, urllib2
from subprocess import Popen
from subprocess import PIPE
from xml.dom.minidom import parseString

SPEEDURL = 'http://IP:8000/' # This will be pyspeed.pypy.org/

data = {
    'commitid': '',
    'project': 'buildbot-test',
    'branch': 'trunk',
    'executable_name': 'profiling',
    'executable_options': 'none',
    'benchmark': '',
    'environment': 'bs-test0',
    'result_value': '',
    'result_date': datetime.today(),
}

""" adding data to codespeed """
def push(data):
    params = urllib.urlencode(data)
    f = None
    response = "None"
    try:
        f = urllib2.urlopen(SPEEDURL + 'result/add/', params)
        response = f.read()
        f.close()
    except urllib2.URLError, e:
        if hasattr(e, 'reason'):
            response = '\n We failed to reach a server\n'
            response += ' Reason: ' + str(e.reason)
```

```

    elif hasattr(e, 'code'):
        response = '\n  The server couldn\'t fulfill the request\n'
        response += '    Error code: ' + str(e)
    print "Server (%s) response: %s\n" % (SPEEDURL, response)

def main():
    a = Test()
    for i in range(1,random.randint(1,5)):
        a.add()
    cmd='svn log --xml'
    pipe = Popen(cmd, shell=True, bufsize=-1, stdout=PIPE).stdout
    output=pipe.read()
    parsed=parseString(output)
    taginfo=parsed.getElementsByTagName('logentry')
    atribut=taginfo[0].getAttributeNode('revision')
    data['commitid'] = atribut.value

    # getting info from resource
    usage = resource.getrusage(resource.RUSAGE_SELF)
    data['benchmark']= 'execution-time'
    data['result_value'] = str(usage.ru_utime)
    push(data)
    # getting info from guppy-pe
    h=hpy()
    data['benchmark'] = 'memoria'
    data['result_value'] = str(h.iso(a.get_buff()).size)
    push(data)

if __name__ == "__main__":
    main()

```

## setup.py

```

from setuptools import setup, find_packages

setup(
    name = "buildbot-test",
    version = "0.1",
    packages = find_packages(),
    entry_points = {
        'console_scripts': [
            'buildbot-test = test.test:main',
        ],
    },
    test_suite = "test.tests",
    include_package_data = True,
    author = "Oriol Rius",
    author_email = "oriol@joor.net",
    description = "simple code for testing buildbot",
)

```

## Apendice II: master.cfg

```
# -*- python -*-

c = BuildmasterConfig = {}

#####
# BUILDSLAVES
#####
# the 'slaves' list defines the set of allowable buildslaves. Each element is
# a BuildSlave object, which is created with bot-name, bot-password. These
# correspond to values given to the buildslave's mktap invocation.
from buildbot.buildslave import BuildSlave
c['slaves'] = [BuildSlave("bs-test0", "test0pass", max_builds=1), BuildSlave("bs-
test1", "test1pass", max_builds=1)]
c['slavePortnum'] = 9989

#####
# CHANGESOURCES
#####
from buildbot.changes.svnpoller import SVNPoller
source_code_svn_url='svn://HOST/path/buildbot-test'
svn_poller = SVNPoller(svnurl=source_code_svn_url, svnuser="buildbot-read",
svnpasswd="x", histmax=10, svnbin='/usr/bin/svn', pollinterval=60)
c['sources'] = [ svn_poller ]

#####
# SCHEDULERS
#####
from buildbot.scheduler import Scheduler
c['schedulers'] = []
c['schedulers'].append(Scheduler(name="sched_tree_stable_timer", branch=None,
treeStableTimer=None,
builderNames=["build-basic"]))

#####
# BUILDERS
#####
from buildbot.process import factory
from buildbot.steps.source import SVN
from buildbot.steps.shell import Compile, Test, ShellCommand
from buildbot.steps.python_twisted import Trial
f1 = factory.BuildFactory()
svnroot = "svn://HOST/path/buildbot-test"
# player-linux/trunk/buildbot-test
f1.addStep(SVN(svnurl=svnroot, directory="build", username="buildbot-read",
password="x", mode='update'))
f1.addStep(Compile(name='compile', description=['compiling'],
descriptionDone=['compile'], command=["python", "setup.py", "build"]))
f1.addStep(Test(name='test', command=["python", "setup.py", "test"],
description=['testing'], descriptionDone=['tested']))
f1.addStep(ShellCommand(name='performance', description='profiling',
descriptionDone='profile', workdir='build/test', command='python profiling.py'))

b1 = {'name': "build-basic",
      'slavename': "bs-test0",
      'builddir': "builddir",
      'factory': f1,
      }
c['builders'] = [b1]

#####
# STATUS TARGETS
#####
c['status'] = []
```

```
from buildbot.status import html
c['status'].append(html.WebStatus(http_port=8010,allowForce=True))

##### DEBUGGING OPTIONS
# client for connect: contrib/debugclient.py - same port as slave builders
c['debugPassword'] = "debugpassword"

#from buildbot import manhole
#c['manhole'] = manhole.PasswordManhole("tcp:9999:interface=127.0.0.1","admin",
#"bb2admin")

##### PROJECT IDENTITY
c['projectName'] = "Buildbot test"
c['projectURL'] = "http://URL"
c['buildbotURL'] = "http://IP:8010/"
```