

# Quodlibot, a Google Code IRC bot

AUTHOR: Steven Robertson  
CONTACT: [steven@strobe.cc](mailto:steven@strobe.cc)  
PUBLISHED: 2009-12-26  
TAGS: Article, Web

I was looking for a simple IRC bot which would announce changes to the [Quod Libet](#) project, like what [CIA](#) does for version control. Finding none, I hacked one together in about an hour, using [Twisted](#) and [Feed Parser](#). It's trivial, but code that isn't shared is lost, and it may save a few others some time in the future. If you maintain a GC project and want me to host an instance of the bot, send me an email.

```
1 # Copyright (c) 2009 Steven Robertson.
2 #
3 # This program is free software; you can redistribute it and/or modify
4 # it under the terms of the GNU General Public License version 2 or
5 # later, as published by the Free Software Foundation.
6
7 NAME="Google_Code_RSS_IRC_Bridge_Bot"
8 VERSION="0.1"
9
10 from twisted.words.protocols import irc
11 from twisted.internet import reactor, protocol, task
12
13 import feedparser
14
15 import re
16 import sys
17 import urllib2
18
19 class AnnounceBot(irc.IRCClient):
20
21     username = "%s-%s" % (NAME, VERSION)
22     sourceURL = "http://strobe.cc/"
23
24     # I am a terrible person.
25     instance = None
26
27     # Intentionally 'None' until we join a channel
28     channel = None
29
30     # Prevent flooding
31     lineRate = 3
32
33     def signedOn(self):
34         self.join(self.factory.channel)
```

```

35     AnnounceBot.instance = self
36
37     def joined(self, channel):
38         self.channel = self.factory.channel
39
40     def left(self, channel):
41         self.channel = None
42
43     def trysay(self, msg):
44         """Attempts to send the given message to the channel."""
45         if self.channel:
46             try:
47                 self.say(self.channel, msg)
48                 return True
49             except: pass
50
51 class AnnounceBotFactory(protocol.ReconnectingClientFactory):
52     protocol = AnnounceBot
53     def __init__(self, channel):
54         self.channel = channel
55
56     def clientConnectionFailed(self, connector, reason):
57         print "connection failed:", reason
58         reactor.stop()
59
60 class FeedReader:
61     _schema = 'http://code.google.com/feeds/p/%s/updates/basic'
62
63     def __init__(self, project):
64         self.project = project
65         self.entries = {}
66
67     def update(self):
68         """Returns list of new items."""
69         feed = feedparser.parse(self._schema % self.project)
70         added = []
71         for entry in feed['entries']:
72             if entry['id'] not in self.entries:
73                 self.entries[entry['id']] = entry
74                 added.append(entry)
75         return added
76
77     def strip_tags(value):
78         return re.sub(r'<[^>]*?>', '', value)
79
80     def announce(feed):
81         new = feed.update()
82         for entry in new:
83             msg = '%s: %s' % (strip_tags(entry['title']), entry['link'])
84             if AnnounceBot.instance:
85                 AnnounceBot.instance.trysay(msg.replace('\n', ' ').encode('utf-8'))
86
87 if __name__ == '__main__':
88     # All per-project customizations should be done here
89
90     AnnounceBot.nickname = 'quodlibot'

```

```
91     fact = AnnounceBotFactory("#quodlibet")
92     feed = FeedReader('quodlibet')
93     reactor.connectTCP('irc.oftc.net', 6667, fact)
94
95     # Don't reannounce every update on startup
96     feed.update()
97
98     update_task = task.LoopingCall(announce, feed)
99     update_task.start(600, now=False)
100
101     reactor.callLater(10, announce, feed)
102
103     reactor.run()
```