

# GOOGLE HACKS™

**3rd  
Edition**  
Covers Google  
Maps

*Tips & Tools for Finding and  
Using the World's Information*



O'REILLY®

**HACKS  
SERIES™**

**Rael Dornfest,  
Paul Bausch & Tara Calishain**

*Foreword by Chris DiBona,  
Open Source Program Manager, Google*

# Google Hacks™, Third Edition

by Rael Dornfest, Paul Bausch, and Tara Calishain

Copyright © 2003, 2005, 2006 O'Reilly Media, Inc. All rights reserved.  
Printed in the United States of America.

Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North,  
Sebastopol, CA 95472.

O'Reilly books may be purchased for educational, business, or sales promotional use. Online editions are also available for most titles (*safari.oreilly.com*). For more information, contact our corporate/institutional sales department: (800) 998-9938 or *corporate@oreilly.com*.

**Editor:** Brian Sawyer  
**Production Editor:** Matt Hutchinson  
**Copyeditor:** Matt Hutchinson  
**Proofreader:** Mary Brady

**Indexer:** Reg Aubry  
**Cover Designer:** Edie Freedman  
**Interior Designer:** Melanie Wang  
**Illustrators:** Robert Romano and  
Jessamyn Read

## Printing History:

February 2003: First Edition.  
December 2004: Second Edition.  
August 2006: Third Edition.

Nutshell Handbook, the Nutshell Handbook logo, and the O'Reilly logo are registered trademarks of O'Reilly Media, Inc. The *Hacks* series designations, *Google Hacks*, the image of locking pliers, and related trade dress are trademarks of O'Reilly Media, Inc.

Google, PageRank, AdSense, AdWords, Gmail, and I'm Feeling Lucky are trademarks of Google Technology, Inc.

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and O'Reilly Media, Inc. was aware of a trademark claim, the designations have been printed in caps or initial caps.

While every precaution has been taken in the preparation of this book, the publisher and authors assume no responsibility for errors or omissions, or for damages resulting from the use of the information contained herein.

*Small print:* The technologies discussed in this publication, the limitations on these technologies that technology and content owners seek to impose, and the laws actually limiting the use of these technologies are constantly changing. Thus, some of the hacks described in this publication may not work, may cause unintended harm to systems on which they are used, or may not be consistent with applicable user agreements. Your use of these hacks is at your own risk, and O'Reilly Media, Inc. disclaims responsibility for any damage or expense resulting from their use. In any event, you should take care that your use of these hacks does not violate any applicable laws, including copyright laws.



This book uses RepKover™, a durable and flexible lay-flat binding.

ISBN: 0-596-52706-3  
[M]

This excerpt is protected by copyright law. It is your responsibility to obtain permissions necessary for any proposed use of this material. Please direct your inquiries to [permissions@oreilly.com](mailto:permissions@oreilly.com).

HACK  
#52

## Google over IM

Build a Google Talk bot that will have you talking directly with Google Search results.

Instant messaging is no longer solely the domain of teenagers with too much spare time on their hands. IM has morphed from a fun toy to a serious productivity tool. Google offers its own IM client called Google Talk (<http://www.google.com/talk/>).

Thanks to the Google Talk and the Google API, you can skip the web site and bring Google Search results directly into your IM client with a Google Talk bot. A *bot* is simply a program that looks like an IM chat buddy (someone who receives and sends messages). Behind the scenes, though, the bot simply does what it's programmed to do. With a Google Talk bot up and running, you can find search results without leaving your IM client, which sounds very productive.

### What You Need

Google Talk uses a standard messaging protocol called Extensible Messaging and Presence Protocol (XMPP). XMPP was developed as part of Jabber, an open IM protocol. Because Jabber has been around for a number of years, there are plenty of existing tools that speak Google Talk's language.

This hack is written in Python and requires the *jabber.py* module (<http://jabberpy.sourceforge.net>) for communicating through Google Talk. The code for talking with the Google API is adapted from the simple Python example [Hack #95] and requires the *pyGoogle* module (<http://pygoogle.sourceforge.net>). And, of course, you'll need a free Google API key, which you can pick up at Google Web APIs (<http://www.google.com/apis/>).

You'll also need a spare Google Account for your bot. Log into Gmail with your Google account and send yourself an invitation. Be sure to log out of Google completely, follow the instructions in your Gmail invite, and sign up for Google using your alternate identity. Jot down the alternate account username and password. Remember that your bot will be logging into Google Talk, so whichever name you give your bot when you sign up will be your bot's identity online.

### The Code

This code provides a bare-bones bot that handles incoming messages and sends simple messages. In addition, the script queries Google and formats the response for instant messages. Be sure to include the login username and

password for your bot. Include your Google API key as well, and then save the following code to a file called *queryBot.py*:

```
#!/usr/bin/python
# queryBot.py
# A Google Talk bot that returns Google Search
# results as messages for any incoming message.
# Usage: python queryBot.py

import sys
import string
import re
import jabber
import xmlstream
import google

username = 'insert google account name' # do not include @gmail.com
password = 'insert google account password'
google.LICENSE_KEY = 'insert google API key'
botname = 'queryBot'

def sendMsg(toid,msg):
    r = jabber.Message(toid,msg)
    r.setType('chat')
    con.send(r)

def messageCB(con,msg):
    if msg.getBody():
        query = msg.getBody()
        fid = msg.getFrom()

        print '>>> query: %s' % query
        print '>>> from: %s' % fid

        # Query Google.
        data = google.doGoogleSearch(query)

        # Output.
        for result in data.results:
            # set the results as variables
            title = result.title
            URL = result.URL
            snippet = result.snippet

            # Strip HTML
            regex = re.compile('<[^>]+>')
            title = regex.sub(r'',title)
            snippet = regex.sub(r'',snippet)
            regex2 = re.compile('&#39;')
            title = regex2.sub(r'',title)
            snippet = regex2.sub(r'',snippet)

            title = '\n*s*' % title # Bold title
```

```
# Format result
response = string.join( (title, snippet, URL), "\n")

# Send result
r = jabber.Message(fid,response)
r.setType('chat')
con.send(r)

def connect():
    global con
    con = jabber.Client(host='gmail.com',debug=[],
                       log='xmpp.log',port=5223,
                       connection=xmlstream.TCP_SSL)

    con.connect()
    con.setMessageHandler(messageCB)
    con.auth(username, password, botname)
    con.requestRoster()
    con.sendInitPresence()
    print "[[ Bot is Online, ready for queries! ]]"

con = None
while 1:
    if not con:
        connect()
    con.process(1)
```

Note that when the Jabber client is initialized in the `connect()` function, a logfile (*xmpp.log*) is set. You'll find a copy of all XMPP messages flying between your machine and the Google Talk server, and it's extremely useful for finding problems with your bot.

## Running the Hack

Before you can run your bot, you'll need a bit of Google login shuffling. To combat spam, Google Talk requires every user to have explicit permission to talk to each other. Log into Google Talk as yourself and send a chat request to your bot's identity. Then log in using your bot's credentials and approve your real self for chatting. You'll need to approve every one your bot chats with.

Also, make sure both *jabber.py* and *google.py* are in the same directory as your script. If they aren't, install them with the *setup.py* scripts that come with the modules.

Once everything is set, open a command prompt and start the script, like so:

```
python queryBot.py
```

The bot should start and give you the opening OK:

```
[[ Bot is Online, ready for queries! ]]
```

At this point, log into Google Talk as yourself and send a simple message to your bot. Back in the command window, you'll see the incoming query and the user who sent it:

```
>>> query: ROTFL
>>> from: user@example.com/Talk.v9222832159
```

At this point, the script takes the incoming message and queries the Google API for search results. As the results come in, they're sent back to the user, as shown in Figure 4-22.

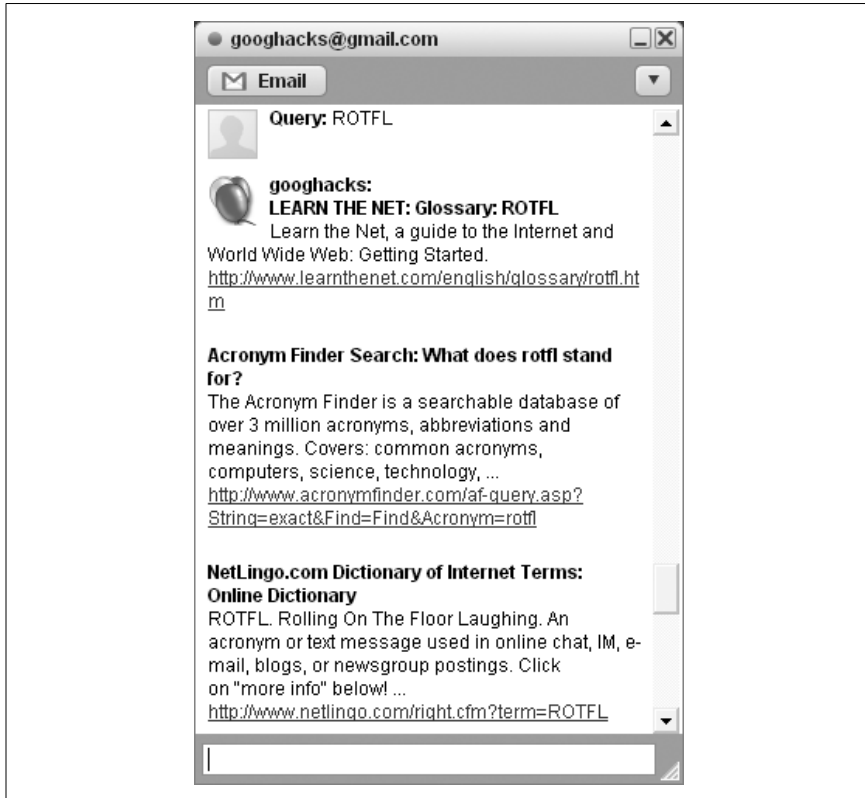


Figure 4-22. QueryBot responding to the message “ROTFL”

The bot sends back the first 10 results for the query as individual messages.

When you're ready to put your bot to sleep, type Ctrl-C in the command window to take your bot offline.

## See Also

- For a fully functioning Google Talk bot that can conference several users together and perform more complex commands, take a look at Google Talk: Conference Bot (<http://coders.meta.net.nz/~perry/jabber/confbot.php>) by Perry Lorier and Limodou—the inspiration for this hack. The code is freely available, and if you're familiar with Python you can customize the bot for your own purposes.
- “Google from IRC” [Hack #50].