

# IRC HACKS™

*100 Industrial-Strength Tips & Tools*



O'REILLY®

*Paul Mutton*

HACK  
#59

## An Egg Timer

Create an IRC bot that can remind you to do something after a set interval.

Some people spend far too long in front of their computers using IRC. If you are one of those people, here is a hack that may appeal to you. Never again will you forget to do something important, because you'll be able to get an IRC bot to remind you!

This hack shows how to create a simple IRC bot that sits in any number of channels and responds to the `!egg` command. After three minutes, the bot will remind you that your egg is done, for this is how long it takes to boil the perfect egg.

Scheduling a task for later execution is easy with Java. The `java.util.Timer` class allows you to schedule tasks that extend `java.util.TimerTask`. The `Timer` class is scalable, so the bot should be perfectly capable of scheduling thousands of tasks at the same time. That's a lot of eggs.

### The Code

You will need to create a special class called `EggTimerTask` that extends `TimerTask`. When the egg is ready, the `run` method in this class will be called. In the `run` method, the bot must send a message to the channel to tell the user that her egg is ready to eat. Instances of this class therefore need to store a reference to the bot, channel, and nickname of the user.

Create the file `EggTimerTask.java`:

```
import java.util.TimerTask;

public class EggTimerTask extends TimerTask {

    private EggTimerBot bot;
    private String nick;
    private String channel;

    public EggTimerTask(EggTimerBot bot, String nick, String channel) {
        this.bot = bot;
        this.nick = nick;
        this.channel = channel;
    }

    public void run() {
        bot.sendMessage(channel, nick + ": Your egg is ready!");
    }

}
```

Writing the actual bot is rather straightforward, as all you need to do is make it respond to messages that look like “!egg”. It needs to create a new `EggTimerTask` and schedule it for running three minutes later.

Create the file *EggTimerBot.java*:

```
import org.jibble.pircbot.*;
import java.util.Timer;

public class EggTimerBot extends PircBot {

    public static final long DURATION = 3 * 60 * 1000;
    private Timer timer = new Timer(true);

    public EggTimerBot(String name) {
        setName(name);
    }

    public void onMessage(String channel, String sender,
        String login, String hostname, String message) {

        if (message.trim().toLowerCase().equals("!egg")) {
            sendMessage(channel, sender + ": I am timing your 3 minutes now.
..");
            EggTimerTask timerTask = new EggTimerTask(this, sender,
channel);
            timer.schedule(timerTask, DURATION);
        }
    }
}
```

Finally, you just need a main method to start the bot and tell it to connect to a server and join a channel. If you want, you can ask this bot to join more than one channel, and it will still happily do its job.

Save the following as *EggTimerBotMain.java*:

```
public class EggTimerBotMain {

    public static void main(String[] args) throws Exception {
        EggTimerBot bot = new EggTimerBot("eggcook");
        bot.setVerbose(true);
        bot.connect("irc.freenode.net");
        bot.joinChannel("#irchacks");
    }
}
```

## Running the Hack

Compile the bot:

```
C:\java\EggTimerBot> javac -classpath .;pircbot.jar *.java
```

Run it like so:

```
C:\java\EggTimerBot> java -classpath .;pircbot.jar EggTimerBotMain
```

## The Results

When someone issues the `!egg` command, the bot will tell him when three minutes has elapsed, as shown in [Figure 9-2](#). Note that multiple timings can run concurrently. If you want, you could even [run this bot in more than one channel](#) [[Hack #50](#)].

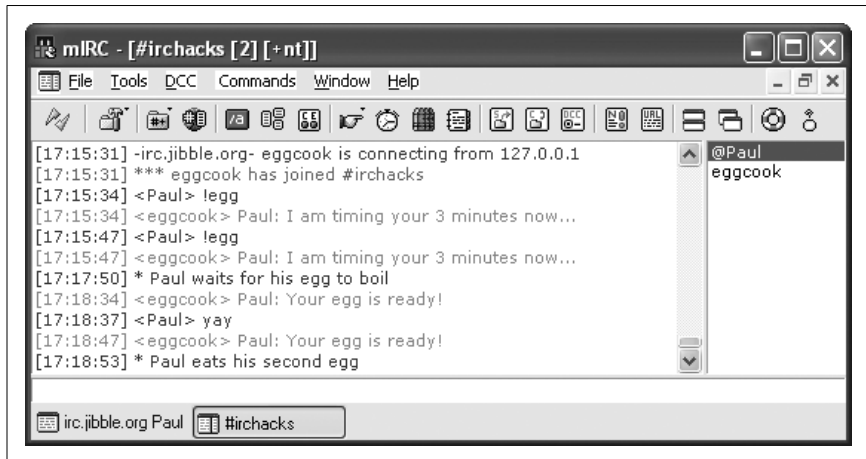


Figure 9-2. The working egg timer

## Hacking the Hack

Timing three minutes is fine if you're a fan of boiled eggs, but not everything takes three minutes to cook. You could generalize the bot so it is suitable for other purposes. Modify the body of the `onMessage` method so it can accept a new command, `!timer duration`, where `duration` is the number of seconds to wait before alerting you:

```
message = message.trim().toLowerCase();
if (message.equals("!egg")) {
    sendMessage(channel, sender + ": I am timing your 3 minutes now...");
    EggTimerTask timerTask = new EggTimerTask(this, sender, channel);
    timer.schedule(timerTask, DURATION);
}
else if (message.startsWith("!timer ")) {
    try {
        int duration = Integer.parseInt(message.substring(7));
        if (duration > 0) {
            EggTimerTask timerTask = new EggTimerTask(this, sender,
channel);
            // Multiply the milliseconds by 1000 to get seconds.
```

```
        timer.schedule(timerTask, duration * 1000);
    }
}
catch (NumberFormatException e) {
    // Do nothing.
}
}
```

You can now get the bot to time any period measured in seconds, for example:

```
!timer 30
```

would make the bot wait half a minute before telling you your egg's ready. Of course, you may not be using it for cooking eggs by this stage, so you may like to change the message that is output by the `EggTimerTask` class.