

# IRC HACKS™

*100 Industrial-Strength Tips & Tools*



O'REILLY®

*Paul Mutton*

HACK  
#35

## IRC with Java and PircBot

Use the PircBot Java IRC API to make bots that connect to IRC.

PircBot is a Java framework for writing IRC bots quickly and easily. It allows you to abstract yourself away from the underlying protocols and concentrate on making bots that do something useful. Using this framework is much easier than implementing all of the grunt work by yourself. Connecting to a server, for example, takes just one line of code.

You can get PircBot from <http://www.jibble.org/pircbot.php>. To make your own bot that uses this framework, you will need to open the PircBot ZIP file and place the *pircbot.jar* file in a new directory. This directory will also be used to store the source code for the bot.

The file *pircbot.jar* contains all of the classes and source code that make up the PircBot package. Curious users will also note that the ZIP file contains full documentation for the package, which is essential for exploring the more advanced capabilities of PircBot. The package contains an abstract class named *PircBot*, which you can extend and inherit from. Creating an IRC bot is a simple case of extending this class, although you may like to change the nickname of the bot from its default of “PircBot.”

Writing a bot becomes much simpler, as illustrated in this second Java version of a simple IRC bot. Save the following code as *HackBot2.java*:

```
import org.jibble.pircbot.*;

public class HackBot2 extends PircBot {

    public HackBot2() {
        this.setName("HackBot2");
    }

}
```

Believe it or not, that’s all you have to do to create a bot. Creating an instance of the *HackBot2* class gives you an object that inherits all of the methods from the *PircBot* class, thereby allowing you to tell it to connect to a server, join a channel, and so on.

Create another class in a new source file, *Main.java*, which contains a simple main method to create an instance of the bot. The main method also tells it to connect to a server and join a channel. A separate class called *Main* is used to store the main method so it is easier to work out how to run the bot—when a newcomer stumbles across your project, it is obvious where the main method is without having to look at any source code.

```
import org.jibble.pircbot.*;
```

```
public class Main {  
  
    public static void main(String[] args) throws Exception {  
  
        // Start the bot up.  
        HackBot2 bot = new HackBot2();  
  
        // Enable debugging output.  
        bot.setVerbose(true);  
  
        // Connect to an IRC server.  
        bot.connect("irc.freenode.net");  
  
        // Join the #irchacks channel.  
        bot.joinChannel("#irchacks");  
  
    }  
  
}
```

The `bot.setVerbose(true)` method call is used to turn on verbose mode. This means that your bot will print out all of the raw commands it receives from the server, as well as printing out the commands the bot itself sends to the server. This is handy for diagnostic purposes if you can't get your bot to work.

## Event-Driven Framework

So far, `HackBot2` will connect to the IRC server and respond to pings from the server, much like the previous `Java HackBot`. You can easily add a few more features because `PircBot` is an event-driven framework. This means you can easily make your bot respond to events as they happen, for example, saying what time it is whenever it is asked. Getting a `PircBot` to respond to such messages is a simple case of overriding the `onMessage` method.

Add this method to the `HackBot2.java` source code:

```
public void onMessage(String channel, String sender,  
                    String login, String hostname, String message) {  
    if (message.equalsIgnoreCase("time")) {  
        String time = new java.util.Date().toString();  
        sendMessage(channel, sender + ": The time is now " + time);  
    }  
}
```

If your bot is in a channel, the preceding method will be called whenever anybody says something. If anybody says, “time,” your bot will respond by telling her what the current time is.

```
<Jibbler> time  
<HackBot2> Jibbler: The time is now Sun Dec 14 13:49:36 GMT 2003
```

As IRC is accessible by people from all over the world, it is a good idea to use this style of presentation, as it also states what time zone the bot is running in.

## Running the Hack

Open up a command prompt and change to the directory that contains *pircbot.jar*, *HackBot2.java*, and *Main.java*. If you are running Windows and have the Java SDK installed and set up correctly, you can compile your bot with the `javac` command. It is necessary to tell it to look in *pircbot.jar* to find the required PircBot classes.

```
C:\java\HackBot2> javac -classpath .;pircbot.jar *.java
```

If you are using Linux or Unix, the command-line parameters will differ slightly, as the `:` character is used to separate paths in the classpath list. You will need to run the following command:

```
% javac -classpath pircbot.jar:. *.java
```

If your bot compiled successfully, without any errors, it should now be ready to run. Your directory will now contain the files *HackBot2.class* and *Main.class*. Again, you must tell the Java runtime where to find the required PircBot classes. Windows users can run the bot with the following command:

```
C:\java\HackBot2> java -classpath pircbot.jar;. Main
```

Once again, Linux and Unix users can run the bot by replacing the `;` with a `:`. The bot should then spring into life and join the specified channel, ready to tell people what time it is.

## Hacking the Hack

The documentation contained in the downloaded PircBot ZIP file is an abundant source of information, with full descriptions of all methods contained in the package. Reading this documentation is a good way to get ideas for features to add to your bot. Because PircBot is suitable for use in Java Applets, you could even use this as a starting point to create a basic IRC client that you can embed into a web page.

You may find inspiration for ideas on the PircBot Implementations page at <http://www.jibble.org/pircbot-implementations.php>. This page lists some of the existing bots and clients based on the PircBot framework. Some of my favorites are:

*ChoonBot* (<http://www.jstott.me.uk/choonbot>)

This provides an IRC interface to Winamp, so everyone in our office can select what music we all listen to.

*PieSpy* (<http://www.jibble.org/piespy>)

This sits in an IRC channel and [infers and visualizes social networks](#) [[Hack #44](#)].

*Azureus* (<http://azureus.sourceforge.net>)

This is a BitTorrent file-sharing client that uses PircBot to run an integrated IRC client, letting file-sharers chat with one another.

*mobibot* (<http://www.thauvin.net/mobitopia/mobibot>)

This bot was designed to capture URLs posted in a channel, and you can find it in [#mobitopia](#) on the freenode IRC network.

*TellyBot* (<http://www.jstott.me.uk/tellybot>)

This is an IRC bot that provides an interface to TV listings information. It allows you to search for programs and can remind you when they are about to start.