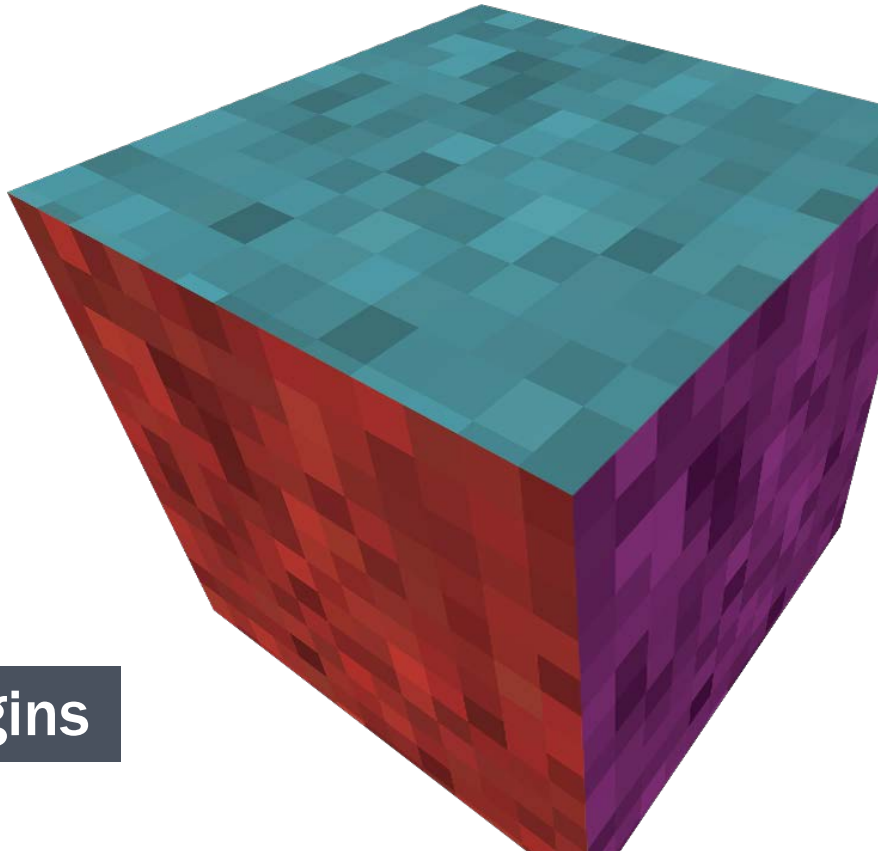
A Beginner's Guide to Writing Minecraft® Plugins in JavaScript

Walter Higgins

A Beginner's Guide to Writing Minecraft® Plugins in JavaScript



Peachpit Press

Walter Higgins

A Beginner's Guide to Writing Minecraft® Plugins in JavaScript

Walter Higgins

Peachpit Press

www.peachpit.com
To report errors, please send a note to errata@peachpit.com

Peachpit Press is a division of Pearson Education.

Editor: Kim Wimpsett
Production editor: Rebecca Chapman-Winter
Compositor: Danielle Foster
Indexer: Valerie Haynes Perry
Cover design: Mimi Heft
Interior design: Mimi Heft

ISBN-13: 978-0-133-93014-6
ISBN-10: 0-133-93014-9

9 8 7 6 5 4 3 2 1

Printed and bound in the United States of America

For my family.

Acknowledgments

I'd like to thank—in no particular order—the many contributors to ScriptCraft (the software used in this book); the teachers and volunteers who have used and improved ScriptCraft; my editors, Cliff Colby and Kim Wimpsett; and the production editor and proofreader for their diligence and enormous help in writing this book. This book took a big chunk of time to write, and I must thank my wife Ursula and kids, Kate and Sean, for their patience, love, and support. I'd also like to thank my dad, Paul Higgins, for instilling in me a love of books.

This book would not be possible without the work of the hundreds of open source programmers who developed or contributed to the open source Bukkit and CanaryMod projects. These projects are the lifeblood of Minecraft and the Minecraft modding community. Thanks to Jason Jones (`@darkdiplomat`) for heroic work maintaining CanaryMod, Michael Madden for creating great ScriptCraft resources for CoderDojo, Athijegannathan Sundararajan for patiently answering all my questions about Nashorn, Chris Cacciatore for his fun ScriptCraft contributions, and Jon Ippolito for his forum contributions.

About the Author

Walter Higgins has more than 20 years' experience in software development working at Microsoft, Apple, EMC, and IBM. When not programming, he enjoys tinkering with new technology, reading, and running. He lives in Cork, Ireland.

Contents

Preface

Who Is This Book For?

This book is for anyone who is curious about programming and creating Minecraft plugins. It teaches how to create Minecraft server plugins and assumes the reader has no previous programming experience. Specifically, this book is for beginning programmers age 10 and older.

Why I Wrote This Book

I've been playing Minecraft since 2010 and have been playing multiplayer Minecraft with my two kids on a shared server at home since 2011. I developed software as a hobby in my teens and have been developing software professionally for more than 20 years. I took my kids to local CoderDojo sessions where they learned to use Scratch and create simple web pages using HTML and JavaScript. I thought "Wouldn't it be cool if kids could learn to program using Minecraft?" When I first looked into developing Minecraft plugins, I was bewildered by the number of options available.

All of the options available required you to write code in Java. Java is the programming language that Minecraft is written in. It is a fine general-purpose language and is especially suited to developing large, complex business applications. However, it is not ideal as a first language to learn. Learning Java can take some time, and you need to write a lot of Java code to get things done. Even for a seasoned Java developer, the options available for modding Minecraft were bewildering. That's why I came up with the idea of making modding easier by letting plugin developers use JavaScript instead.

In late 2012 I launched ScriptCraft—a way of writing Minecraft plugins using JavaScript, which is a much simpler language than Java.

I wrote this book to make learning to program fun and easy. I believe that learning even a little about software—how it's made and how it works—is good. Developing Minecraft plugins is a great way to learn programming and create something fun for yourself and your friends. Maybe you want to create your own Minecraft mini-game or you can't find a plugin that does exactly what you want. This book will teach you how to create your own plugins and mini-games. Playing Minecraft is fun. Creating Minecraft plugins is a whole different level of fun.

Walter Higgins, April 2014

Introduction

This book shows you how to create your own Minecraft server plugins using JavaScript. There's often confusion about the words *mod* and *plugin*. For the purpose of this book, they mean the same thing. Mojang—the makers of Minecraft—prefers to use the term *plugin API* rather than *modding API* when referring to its forthcoming official API, which will make extending Minecraft easier.

Before I begin, I will explain some of the words I'll use throughout this book.

- **Plugin**: A modification you add to Minecraft that changes the game in some way. The plugin is usually in the form of one or more files.
- **Mod**: Mod is short for "modification." In this book, *mod* and *plugin* are used interchangeably (they mean the same thing).
- **Modding**: The practice of writing modifications or plugins for Minecraft. Modding requires some programming knowledge, which you will learn in this book.
- **API**: API is short for "application programming interface," which is an official way to write Minecraft plugins using a guide. Players and regular users of software don't need to care about APIs, but they are essential for programmers because they make it easier to develop plugins. An API is like a list of recipes; you probably crafted your first pick-axe by referring to an online guide—how much more difficult would it be to have tried creating one without knowing where all the materials should go in the crafting grid? Similarly, programmers need APIs to provide help and guidance in building plugins. The API you will use in this book is the CanaryMod API.
- **Multiplayer**: The mode of playing Minecraft with other players all connected to the same server.
- **Server**: A computer that is running the Minecraft server software. In this book I will show you how to set up and run your own Minecraft server.

Why JavaScript?

In this book I will teach you how to program using the JavaScript programming language. JavaScript is just one of many programming languages. When I set out with the goal of making Minecraft plugin development easier, I chose JavaScript for a few reasons.

- JavaScript is an easy language to learn.
- I like JavaScript. I enjoy writing code in JavaScript and hope you will too.
- JavaScript is expressive. You can do more with less. A little bit of JavaScript code can do quite a lot compared to other programming languages such as Java. The shorter your programs are, the easier they are to understand.
- JavaScript is bundled with Java. The latest versions of Java include JavaScript. This means you won't need to install any additional software to use ScriptCraft.
- Anything you can do using Java, you can do using JavaScript. In this book you'll learn about the CanaryMod API—a set of guidelines for creating Minecraft plugins. The CanaryMod API is Java-based, but you can use JavaScript too!
- JavaScript isn't a toy. JavaScript is a proper programming language used professionally by thousands of programmers around the world. Although originally used only for adding simple extensions to web pages, it's now used for developing all kinds of software and has become one of the most popular programming languages.
- JavaScript is cross-platform. This means JavaScript is available on Macintosh, Windows, and Linux.

The JavaScript programming language is a simple language to get started with, but it can be quite flexible in how you do things. In JavaScript there's usually more than one way to do it. This book is not intended as a comprehensive tour of all of JavaScript's features. Its aim is to focus on the fun parts of JavaScript and to help those who are curious about programming get a taste for what programming in JavaScript is like.

About the Upcoming Minecraft API

When I created ScriptCraft in 2012, there was no one true way to write Minecraft plugins because there was no official API provided by Mojang. At the time of writing (September 2014), Mojang has announced an upcoming official way to write Minecraft plugins using the Plugin API. The Plugin API will provide a standard way for plugin developers to extend the Minecraft game. There is still no official release date for the Plugin API, but I'll be watching closely and will make any necessary changes to ScriptCraft when the API is released.

Online Bonus Appendixes

We have provided some additional bonus material in the form of five appendixes, which you can find online at `www.peachpit.com`. They cover the variables provided by ScriptCraft, working with plug-ins, and more. Visit peachpit.com/register, log in or create an account, and register the book's ISBN to download companion files.

Part I

Building a Modding Workbench

WELCOME TO THE EXCITING WORLD of Minecraft plugin development. In this first part of the book, I'll introduce you to the tools you'll need to start programming. In Minecraft, you need to first create a *workbench* (also sometimes called a *crafting table*) before you can craft more sophisticated tools. So too in real life you'll need to first gather some resources and set up a work area where you can create sophisticated Minecraft plugins. This part of the book describes the resources you'll need, how to get them, and how to set up your "modding workbench." All of the resources you'll need are freely available online. The ingredients you'll need to construct your modding workbench are as follows:

- **CanaryMod:** A freely available open source Minecraft server
- **ScriptCraft:** A plugin for CanaryMod
- **A text editor:** gedit, TextWrangler, or any other text editor suitable for programming

The following chapters will show you how to install and set up each of these.

This page intentionally left blank

Chapter 1

Running Your Own Minecraft Server

I BEGAN PLAYING MINECRAFT in 2010 in single-player mode. It was a fun and relaxing way to pass the time. Later I installed the Minecraft server software on one of the computers on our kitchen table. My youngest son, Sean, had just finished his homework, so I asked him if he wanted to join the home server. Seeing each other in the game was quite a novelty. Soon my eldest daughter, Kate, joined in, and we began some serious building together (FIGURE 1.1). Building in Minecraft is much more fun with others.

FIGURE 1.1 Multiplayer Minecraft

When you play Minecraft in multiplayer mode, chatting, building, and mining with other players, you do so on a Minecraft *server*. The Minecraft server is just a computer program (like the Minecraft client, Microsoft Word, or Internet Explorer) that provides a shared virtual place for Minecraft players to connect, build, and chat. You don't need to buy any special type of computer to run a Minecraft server. You can run one on your laptop computer—assuming it's not too old. Minecraft server software won't run on an iPad, Android tablet, or phone. It runs only on Windows, Macintosh, and Linux computers.

Client-Server Networking

Minecraft multiplayer is "client-server," which means that one or more "client" computers (computers running the Minecraft game) can connect to a central "server" (a computer that is responsible for storing all of the Minecraft world information so that players can play together in the same world). **FIGURE 1.2** shows a simple client-server network with just one client (player) connected to a server.

The client must be connected to the server over some kind of network. The line between the client and the server in Figure 1.2 represents the network. A server with just one client computer isn't much fun, though. Servers really become fun when many players are connected to the same server, as in **FIGURE 1.3**.

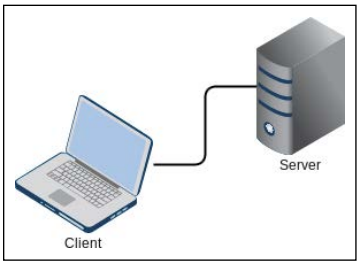


FIGURE 1.2 A simple client-server network

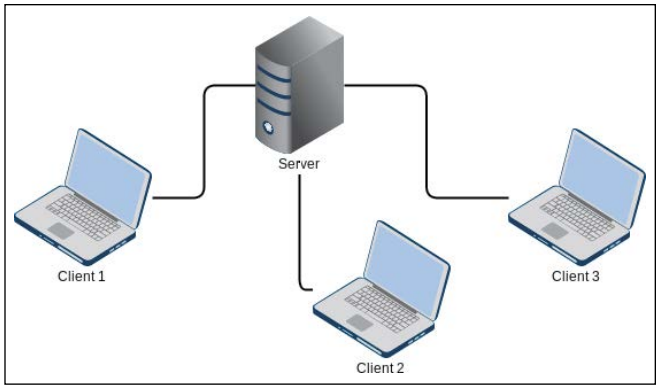


FIGURE 1.3 A client-server network with many clients

In Figure 1.3, the server must chat with all of the connected clients. For example, if it starts raining in the game, the server must send a message to each client that it's begun raining. In Minecraft multiplayer, the server is responsible for many things. The server decides what the weather will be like in the in-game world, what time of day it is, and so on. This is so that the time of day and weather *will be the same for all connected players*.

You'll notice that in Figure 1.3, the server is shown as a separate computer. This is just for illustration. While often server software *does* run on a separate computers, you won't need a separate computer to run Minecraft server or to follow the tutorials in this book. In fact, for learning how to create Minecraft plugins and to get the best from this book, I recommend running a Minecraft server on the same computer you play Minecraft on. This means your computer will act as both a client and a server (see **FIGURE 1.4**). This is by far the most convenient and cost-effective way to run a server and learn how to create Minecraft server plugins.

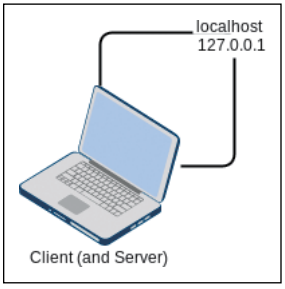


FIGURE 1.4 A computer that is both a client and a server

When you play Minecraft in multiplayer mode, before you can connect to a server, you have to enter the server details into the game. The server details are a hostname or an IP address; either one of these provides a way for the Minecraft client (the game) to connect to a computer running the Minecraft server. The hostname or IP address are how computers uniquely identify one another on the Internet, much like how you identify each other by name and/or address. When you run a Minecraft server on your own computer, you will use a special hostname called localhost or IP address 127.0.0.1. I'll talk more about these in the following sections.

Anyone can run a server. In the following sections, I'll show you how to set up and run your own Minecraft server.

Why Run Your Own Server?

Why should you run your own server? Running your own server means you have total control over the Minecraft world you want to create and share with your friends. You can decide what type of server and world you and your friends will play in, what server plugins to install, what rules will govern the server and world, and who can and cannot join the server. As a server *operator*, you will have access to commands that normal players don't. You can control the weather using Minecraft commands, start and stop thunderstorms, or even change the time of day. You can also install popular Minecraft server mods like Hunger Games and MobArena. The real power of running your own server comes when you add ScriptCraft, which makes it easier to create your own Minecraft server plugins. If you have an idea for a new Minecraft mod or would like to do something that current mods don't support, you can create your own mod and test it on your own server.

Running your own server is essential for plugin development. Minecraft server plugin developers run their own servers because they need to be able to test their creations before releasing them into the wild for others to use. Serious modders would not consider releasing a plugin that they hadn't first tested on their own servers.

Which Minecraft Server Software Should I Use?

There are a couple of different flavors of Minecraft server. Mojang provides its own server software that can be downloaded from the Minecraft.net website. This was the first server software released for Minecraft. This is often referred to as the "vanilla" Minecraft server. At the time of writing, it doesn't provide any way to add plugins, although this will probably change once Mojang releases the official Plugin API. The Minecraft server provided by Mojang is fine if you want to run a server with no modifications. The problems with running such a server is that there's no reliable way to protect against *griefing*.

TERM Griefing

Griefing is when a player in a multiplayer video game deliberately irritates and harasses other players within the game, using aspects of the game in unintended ways. A griefer enjoys annoying other users by destroying other players' work or cursing and harassing players and server administrators.

The standard (vanilla) Minecraft server has only limited ways to protect against griefing. Because of this, other more popular Minecraft server software arose that allowed administrators to strictly control who could join the server and to enforce penalties and bans on players who misbehaved. CanaryMod, which you'll use in this book, let administrators add any number of antigriefing plugins to make the server as safe, secure, and player-friendly as possible.

CanaryMod

In this book, you'll use CanaryMod as your Minecraft server software. You'll look at the CanaryMod API in more detail later. For now, all you need to know is that CanaryMod is the name of the server software, and it has an API that makes it easy to modify the game.

I highly recommend browsing the CanaryMod website at *www.canarymod.net/*. It provides a wealth of information for both administrators (the people who run CanaryMod servers) and developers (the people who create plugins for CanaryMod).

Installing CanaryMod

The first step in constructing your modding workbench is to install CanaryMod. Unlike many games, Minecraft is in ongoing development. This means new versions of Minecraft are released often. When you purchase and download the

TERM Bug

Bugs in software are errors or mistakes in the software code that can cause problems. Nobody quite knows for sure why errors in software are called bugs, but one story goes that a problem with an early mechanical computer in the 1940s was caused by a moth that somehow found its way inside. The term bug had been used to describe errors long before computers came along, so when the engineers captured the moth, they kept it with a note that said "First actual case of bug being found."

Minecraft client, you automatically get new versions of the game when they are released. This is great for players because you get the latest and greatest version for free, which often includes bug fixes and new features. If you've played Minecraft multiplayer before on one of the popular public servers, you'll know that there's a downside to Mojang's commitment to improving Minecraft: Server software upgrades do not happen as often as client software upgrades. This can lead to client software and server software that are *incompatible*. When the client and server versions are incompatible, they cannot communicate (see **FIGURE 1.5**).

Usually, a compatible version of the Minecraft server software is released shortly after the Minecraft client is upgraded. One thing to keep in mind is that CanaryMod is an open source and voluntary project—this means that the people who develop CanaryMod don't get paid to do so. They do it for fun. Upgrading server software takes a lot of work, and the software needs to be tested (by volunteers) before it can be released. That's why sometimes it may take a while for a new version of the Minecraft server software to be released that is compatible with the latest and greatest client. Be patient and remember that the developers who work on CanaryMod do it for fun and don't get paid, so posting questions on the CanaryMod forums, asking when the new version will be ready, won't speed things up.

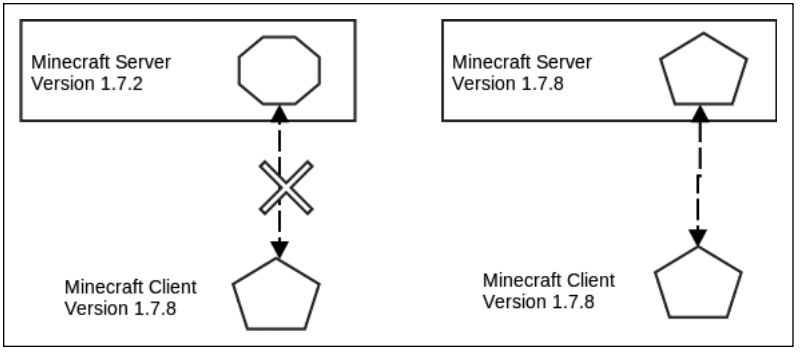


FIGURE 1.5 Compatibility between client and server

At the time of writing, 1.8 is the most current version of Minecraft, and the CanaryMod development team are working to support version 1.8, but the most stable version of CanaryMod (at the time of writing) is version 1.7.10. The example code in this book is tested against CanaryMod version 1.7.10, so I strongly recommend downloading and installing that particular version. There's a way you can safely and temporarily *downgrade* your Minecraft client software so that it will be compatible with your server software.

Since version 1.6, the Minecraft Launcher (the software you run on your computer to start the Minecraft client and to upgrade to new versions when they become available) lets you choose which version of the client software to run. By default the Launcher is configured to always download and run the latest version, but you can easily change this using the new profiles feature. The following are step-by-step instructions for changing your Launcher profile to use a different version of Minecraft client:

1. Launch Minecraft.
2. Click the New Profile button on the bottom left of the Launcher screen (see **FIGURE 1.6**).
3. Fill in the Profile Name field. For example, you should call it something like Version 1.7.10. I find it helpful to include the version name in a new profile so I can see at a glance which client version the profile will use.

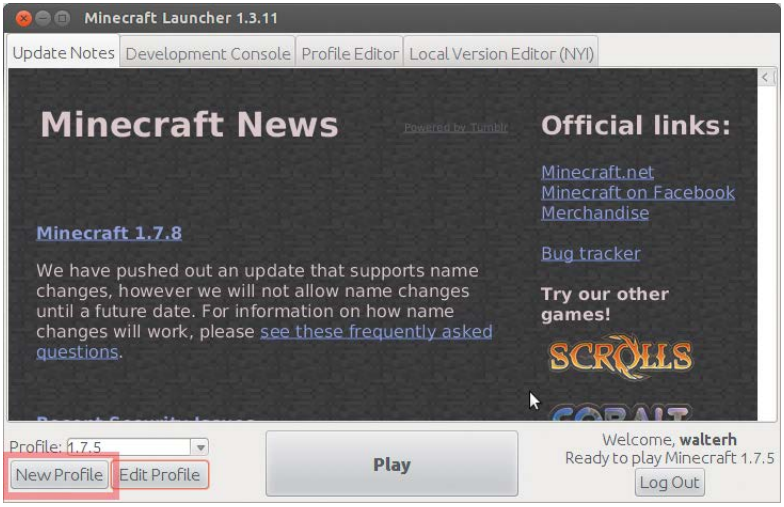


FIGURE 1.6 Minecraft Launcher

4. Choose the appropriate version from the Use Version drop-down list (see **FIGURE 1.7**).
5. Click the Save Profile button.

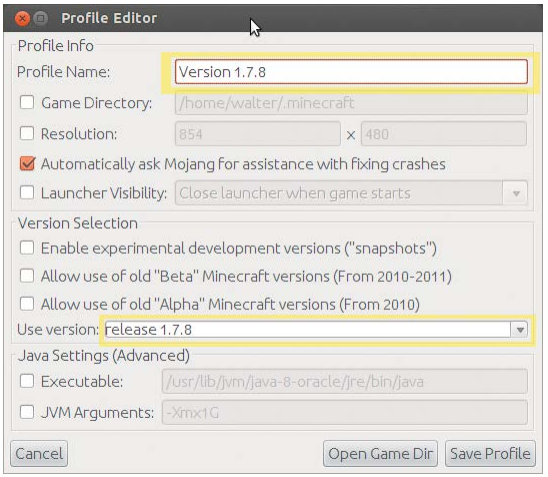


FIGURE 1.7 Minecraft Profile Editor

The newly created profile should now be automatically selected in the Profile drop-down list on the main Launcher page. Click the Play button to begin playing using the chosen version of Minecraft client software (see **FIGURE 1.8**).

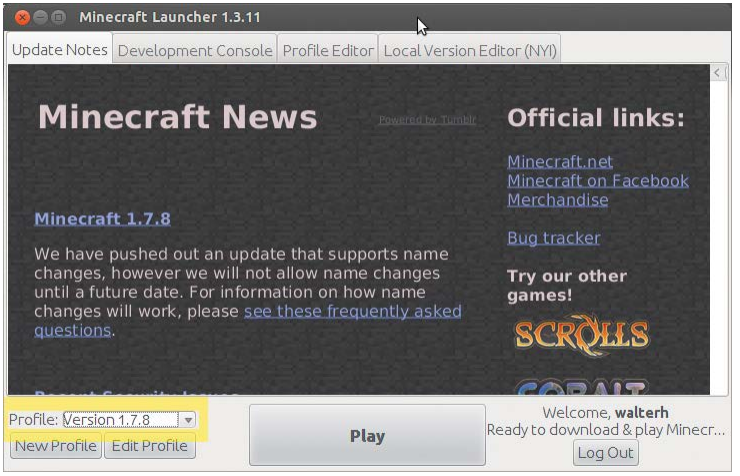


FIGURE 1.8 Profile selection

By using the Profile Editor to downgrade your Minecraft client software, you can ensure that your client and server are compatible and that you can play multiplayer Minecraft even when the server software is not as up-to-date as the current latest version of Minecraft client. This means you can always use the most stable recommended build of CanaryMod.

Depending on which browser software you use (Chrome, Safari, Internet Explorer, or Firefox), the instructions for downloading will differ. Please read the instructions for the browser you use.

TERM JAR File

The CanaryMod software is downloadable as a JAR file. Programs that are created using the Java programming language are often distributed as JAR files and will have a name that ends with `.jar`. JAR is short for **J**ava **AR**chive file. It's an archive of all the code and resources needed by Java to run the software. CanaryMod is distributed as a JAR file. So too are all of the plugins for CanaryMod.

Download Instructions

Follow these steps to download CanaryMod:

1. Create a new folder on your computer called mcserver.
2. Visit *http://scriptcraftjs.org/download/latest/*. Right-click the `canarymod.jar` link and choose "Save link as."
3. Save the JAR file in the folder created in step 1.

Once downloaded, the next step is to install CanaryMod on your computer. The instructions are slightly different for each operating system, so skip directly to the instructions for your operating system.

Installing CanaryMod on Windows

Follow these steps to install CanaryMod on Windows:

1. Open Notepad (you can find it by clicking the Start or Windows button in the bottom left of the screen and then typing `Notepad` in the search box).
2. Type the following into Notepad:

```
java -Xmx1024M -jar canarymod.jar -o true
PAUSE
```

3. Save the document as `run.bat` in the `mcserver` folder you created earlier. It's important that you change the Save As Type option from Text Documents to All Files. Otherwise, Notepad will try to save the document as `run.bat.txt`, and it will not be executable.

4. Double-click the **`run.bat`** file you saved in the previous step, and CanaryMod should start running. If you see the error message "Java is not recognized as an internal or external command, operable program or batch file," then you need to reinstall Java. Java is the programming language Minecraft is written in. It's also the programming language the CanaryMod Minecraft server is written in. You can download Java by visiting *www.java.com* and following the instructions. If after installing Java you still get the same error, then follow the instructions at *www.java.com/en/download/help/path.xml*.
5. Assuming the server has started (you didn't see any errors), you can shut down the server by issuing the **`stop`** command at the server console (type **`stop`** and then hit Enter).

Installing CanaryMod on Mac OS

Follow these steps to install CanaryMod on Mac OS:

1. The first thing you'll need to do is install Java. Since OS X 10.7, Java no longer comes bundled with OS X, so you will need to install Java from Apple's Java Install page at *http://support.apple.com/kb/DL1421*.
2. Open TextEdit. (You can find it by typing **`TextEdit`** in the Spotlight field in the top right of the screen. Click the magnifying glass icon to bring up Spotlight.)

   Under the Format menu, choose Make Plain Text and then type the following:

   ```
   #!/bin/bash
   cd "$( dirname "$0" )"
   java -Xmx1024M -jar canarymod.jar -o true
   Save the file in your mcserver folder as
   start_server.command.
   ```

3. You will need to be able to run this newly created file, so follow these steps:
   a. Open the Terminal application. (To find it, click the magnifying glass icon and type **`Terminal`** in the Spotlight field.)
   b. Type **`chmod a+x`** (don't hit Enter yet).
   c. Find the file you just created in the Finder and drag and drop it into the Terminal window.
   d. The name of the file you just dropped into the Terminal window will be appended to the **`chmod a+x`** command, so you can now hit Enter.

4. Start the server by double-clicking the **start_server.command** file.
5. To stop the server, issue the **stop** command in the Terminal window (type **stop** and then hit Enter).

Installing CanaryMod on Linux (Ubuntu)

Follow these steps to install CanaryMod on Linux:

1. Open Text Editor by clicking the Dash Home button in the top-left corner of the screen; then type **Edit** to see a list of editors (you can choose Text Editor or another editor of your choice).
2. Type the following into the editor:

```
#!/bin/sh
BINDIR=$(dirname "$(readlink -fn "$0")")
cd "$BINDIR"
java -Xmx1024M -jar canarymod.jar -o true
```

3. Save the file as **canarymod.sh** in the **mcserver** folder you created earlier.
4. You will need to make this new file executable. In Nautilus (Linux's File browser) right-click the file and select Properties. In the Properties dialog, click the Permissions tab and ensure that the Execute checkbox is ticked. (You can also make the file executable by issuing the Unix command **chmodx canarymod.sh`** from the folder where you saved the file.)
5. Launch the Terminal application and type **~/mcserver/canarymod.sh** (assuming you created a folder called **mcserver** in your home folder).
6. Once the CanaryMod server console has launched, you can stop it by issuing the **stop** command (type **stop** and then hit Enter).

Agreeing to the End User License Agreement

As of version 1.7.10 of the Minecraft server, you are required to agree to Mojang's end-user license agreement before running the Minecraft server for the first time. The first time you run CanaryMod, the server will stop immediately and issue the following error message: "You need to agree to the EULA in order to run the server."

To agree to the EULA, open the **eula.txt** file in the **mcserver** folder and change it so it reads as follows:

```
eula=true
```

Save the **eula.txt** file and the start CanaryMod again. This time it should run without error.

Configuring Your Server

Once you've installed CanaryMod, the next step is configuration. This is where you decide what kind of Minecraft server you want to run. Will it have monsters? Will it be Survival mode or Creative mode? Will it be player versus player, and so on? While configuration is something you'll probably do only once, you can later change the server configuration at any time you like. The server configuration isn't set in stone; just bear in mind you will need to restart your server if you change the configuration.

The server.cfg File

When you first launch CanaryMod, a couple of configuration files and folders will be created in the **mcserver/config** folder. When configuring your server, the most important file is **server.cfg**. This is a plain-text file (meaning it can be edited using any text editor) containing all of the configuration for your server. The first few lines of my own **server.cfg** look like this:

```
#Minecraft server properties
#Sun Apr 20 11:11:20 IST 2014
op-permission-level=4
allow-nether=false
allow-flight=false
announce-player-achievements=true
server-port=25565
```

The first two lines—the lines that begin with a **#** character (also known as the *hash symbol*)—are comments, so they are ignored by the server. The configuration begins properly on line 3. The settings for the server are specified using *name-value* pairs, which is programmer-speak for how settings are written. For example, **allow-flight=false** is a name-value pair where the name is **allow-flight** and the value is **false**. In property files, the name and value are

separated by the = (equal) symbol. Let's look at just some of the properties you might want to change.

World Type

Later in the book I'll show you how to use JavaScript to create roads and skyscrapers. If you plan to eventually put your newfound programming prowess to use building large-scale places for others to visit, you'll probably want to edit the `mcserver/config/worlds/default/default_NORMAL.cfg` file and change the `world-type` property from `DEFAULT` (Minecraft's standard setting where worlds are made of varied landscapes of hills, valleys, and oceans) to `FLAT`. In a `FLAT` world, there are no hills and no valleys, just flat terrain for as far as you can see. While this might be considered boring, it does have the advantage of being easier to build on. If you plan to build roads, houses, and other large-scale constructions, a `FLAT` world might be the best place to start. It's not essential that `world-type` is set to `FLAT`; you can always leave this setting at `DEFAULT` and just flatten the area you want to build on first. (If you later decide to change the level type, the setting applies only when creating new worlds. Existing worlds will not be modified.)

If you decide to create a new world and want to use a level type other than `DEFAULT`, you have two options.

- Change both the `world-name` and `world-type` properties to create a new world while keeping the old world. (The old world will not be wiped out—it will still be on your computer stored in the `worlds` folder.) This is the best option if you want to keep your existing world data. You need to change the `world-name` setting to something other than the default value. Call your new world something memorable like `flat-world`.
- If you don't care about keeping the existing world, delete the `worlds` folder from the `mcserver` folder and change the `world-type` property. The next time the server starts, a new world and a new `worlds` folder will be created.

Monsters

The next setting you'll probably want to change is the `spawn-monsters` setting in the world's config file located in the `mcserver/config/worlds/` folder. This determines whether monsters will be spawned. By default it's set to `true`, which means monsters will appear at night. I recommend changing this setting from `true` to `false` because the last thing you need while learning to program is to

be attacked by giant spiders. You can change this setting any time you like. The setting won't take effect until you restart the server. I'll talk about starting and stopping the server shortly.

Game Mode

For the same reason I suggest making your server a monster-free zone, I also suggest changing your server's **gamemode** property from 0 (Survival) to 1 (Creative). In Survival mode, you will need to constantly seek food and be careful not to fall. In Creative mode, you can give your full attention to learning to program.

The ops.cfg File

The **ops.cfg** file is a configuration file listing all of the *operators* for the server. Operators are players who have special privileges on the server. Operators (also sometimes called *administrators* or *admins*) are responsible for keeping the server running and ensuring that players play nice and don't grief. If players don't play nice, operators have to power to ban or kick players off the server using special commands available only to them.

There may already be an **ops.cfg** file present in your **mcserver/config** folder. If there isn't, then one will be created automatically when you run the **op** command from the server console. You will need to be an operator to issue JavaScript commands in the game, so one of the first things you should do once you've installed the Minecraft server is make yourself an operator by issuing the **op username** command, replacing **username** with your own Minecraft username. For example, I would issue the command **op walterh** because **walterh** is my Minecraft username. ScriptCraft—the plugin you'll install later—allows only operators to issue JavaScript commands.

You're done with all of the server configuration you'll need for now. If you want to delve deeper into server configuration, a good place to start is by reading the *http://minecraft.gamepedia.com/Server.properties* page.

Permissions

CanaryMod provides a default set of groups: visitors, players, mods, and admins. Players who join a CanaryMod server are assigned to the **visitors** group by default. The **visitors** group can't place or break blocks by default, so you'll need to run the following command once:

```
groupmod permission add visitors canary.world.build
```

You can find out more about CanaryMod permissions by visiting the following links:

- *http://canarymod.net/books/canarymod-admin-guide/permissions-and-groups*
- *http://canarymod.net/books/canarymod-admin-guide/list-permissions*

Basic Server Administration Commands

CanaryMod's usefulness as a Minecraft server comes from its ability to be extended and made more interesting by adding plugins. In this book you'll use just one plugin—ScriptCraft—which lets operators extend the game using JavaScript. There are just a few things you'll need to know about server administration to get the most from this book.

Starting and Stopping Your Server

To start your Minecraft server, double-click the startup script you created earlier in this chapter. This launches the Minecraft server in a Terminal window. The Terminal window will look something like **FIGURE 1.9**.

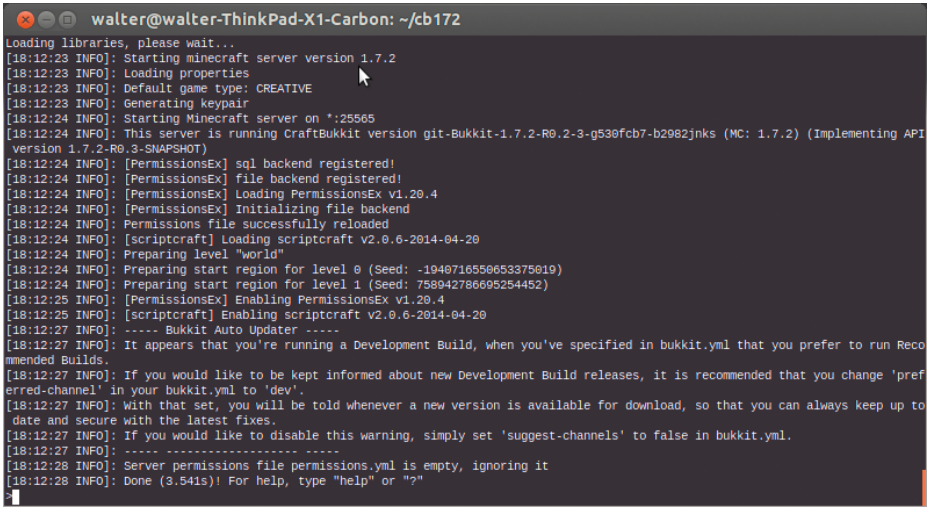


FIGURE 1.9 The Minecraft server console window

Don't worry if your Terminal window doesn't look exactly like this or has slightly different content. The important thing is that after starting up, your server should display a server console prompt—the right arrow (**>**) symbol—in the bottom left of the screen along with a blinking cursor. This is called the *server console*, and you can issue administration commands here even if you aren't an operator. Try it: issue the **`help`** command to see the full list of commands you can use as the server console user.

In the server console window, you don't need to put a forward slash (**/**) in front of commands the way you do in the Minecraft client. So, you simply type the command name without the leading **/**. Remember: **`help`** instead of **`/help`**, **`toggledownfall`** instead of **`/toggledownfall`**, and so on.

Another thing to note about the server console is that when you enter commands there, you do so as a special player called CONSOLE. In Minecraft the **`/me`** command is used to send a message to other players in the form of *yourname action*. For example, if *I* issue the command **`/me sneezes`** in the Minecraft client, all other players on the server will see **`walterh sneezes`**. However, if you issue the same command in the server console, you will see **`* @ sneezes`** instead of your own name. CONSOLE is not visible to other players and does not inhabit the Minecraft world. Normally only the person who starts the server can issue commands as CONSOLE.

To stop the server, you should issue the **`stop`** command at the server console. If you plan to administer a Minecraft server for others to play, the **`stop`** command will come in handy whenever you want to shut down the server for maintenance or troubleshooting.

Connecting to Your Server

To check that your server is running and accepting connections, follow these steps:

1. Launch the Minecraft game as you would normally.
2. Once Minecraft has launched and you have logged in, click the Multiplayer button.
3. If this is the first time you're connecting to your own server, click the Add Server button. If it's not the first time you've connected to your own server, skip to step 8.

4. Fill in the Server Name field. You can call the server any name you like. The default name, Minecraft Server, will do if you can't think of something imaginative right now.
5. In the Server Address field, you should type **`localhost`**. As mentioned earlier, localhost is a special server address meaning "the same computer." You could also use 127.0.0.1, which is the IP address of localhost (servers usually have a name and/or address, and it's usually possible to connect to them using either name or address). Use either localhost *or* 127.0.0.1 but not both; either one will do fine.
6. Once you've entered the server address, click the Done button.
7. The new server you just added should appear in the list of servers. If it's not visible, use the scroll bar to scroll down. If you have already added other servers, it may not appear at the top of the list. Because you've added this server using the Add Server button rather than the Direct Connect button, it should appear in the list of multiplayer servers every time you launch Minecraft.
8. Select the server and click the Join Server button.

If you get an error when trying to connect, ensure that your server is first running (see the earlier section on starting your server). If your server is not running, you will see an error "Failed to connect to the server. java.net.ConnectException: Connection refused." If you see this error, click the Back to Title Screen button; then start your server and try again.

Once you've logged into your server, take a look around. It's important that you have operator (administrator) privileges on your own server. You can check that you have the right privileges by issuing the **`/time set 4000`** command. If you can run this command without error, then you're all set up. If you see a warning message "You do not have permission to perform this command," then you'll need to make yourself an operator at the server console window. Switch to the server console window (Alt+Tab on Windows, Cmd+Tab on OS X) and type **`op username`**, replacing **`username`** with your own username; then switch back to the Minecraft game and try issuing the **`/time set 4000`** command from the in-game prompt. Once you've verified you have operator permissions, kick back and relax. Or, if you have a spare computer on your home network, invite a friend or family member to join you on your server. If this is the first time you've played Minecraft multiplayer with friends or family, you'll have great fun.

Achievement Unlocked!

By now you should have your very own Minecraft server installed and running. Congratulations, you're well on your way to becoming a Minecraft modder!

Other Server Options

Other Minecraft server options are available, but installing and running your own server is the only sensible option if you want to start modding.

Commercial Minecraft Hosting

Once you've mastered server administration and modding, you might eventually decide to use one of the many commercial Minecraft hosting plans available online. These Minecraft hosting providers usually charge a monthly fee, so they are not free. For your money, they provide you with a Minecraft server you and your friends can connect to and play. They usually offer managed install of plugins through a web-based administration portal (a web page where you can choose which plugins your server should use). Commercial Minecraft hosting isn't cheap, and it's not as flexible as running your own server. You certainly don't need to sign up to commercial Minecraft hosting to get the most from this book. You can find a list of Minecraft hosting providers by searching for *Minecraft hosting* online.

Minecraft Realms

Mojang has rolled out its own Minecraft hosting solution—Minecraft Realms—throughout the world. Minecraft Realms does not currently support plugins of any kind, although that may change in the future.

Chapter 2

Setting Up ScriptCraft

IN THE PREVIOUS CHAPTER you downloaded and installed CanaryMod, a customizable Minecraft server. In this chapter, you'll install ScriptCraft—a server plugin that lets you write your own plugins using the JavaScript programming language.

What Is ScriptCraft?

ScriptCraft is a plugin for Minecraft that lets you create plugins using JavaScript instead of Java. It is a server plugin, which means it is installed on a server (CanaryMod) but once installed can be used by all players connected to that server. I created ScriptCraft for several reasons:

- To make it easier for myself to create Minecraft mods
- To make it easier for my kids (and others) to create Minecraft mods
- To make it easier to teach programming to kids using JavaScript and Minecraft as tools

Most Minecraft plugins are written in a language called Java. Java was invented in the 1990s as a simpler way to program. Before Java, programmers used languages such as C and C++, which were difficult to learn and use. C and C++ were difficult because if you wrote a program that wanted to grab some space in the computer's memory, you had to remember to free up that space when you no longer needed it. Otherwise, the computer would quickly run out of memory, and the program would crash. Java solved this problem by having automatic *garbage collection* (yes, that's what programmers actually call it), which automatically frees up memory when it is no longer needed. Java has many other advantages too. It has a large library of functions to do common tasks so you don't have to write those functions yourself. Java is *cross-platform*, which mean it runs on Windows, Linux, and Mac OS. Minecraft is written in Java and so too are Minecraft servers.

Java is a fine language and is widely used. However, it can be a little difficult to learn, especially if you are completely new to programming. Java can be verbose, meaning you need to write a lot of Java code to do even simple things. Java code can't be executed right away either. You need to first *compile* it. Compiling is the process of converting Java source code into a form the computer can understand.

JavaScript came along shortly after Java. It was invented in the mid-1990s just when the Internet was becoming really popular. Although their names sound alike, they are very different languages. JavaScript is simpler than Java in many ways. In recent years, it has become popular as a language not just for web programming but for all kinds of uses.

Installing ScriptCraft

To download and install ScriptCraft, follow these steps:

1. Visit *http://scriptcraftjs.org/download/latest/*.
2. Right-click the **`scriptcraft.jar`** file and select Save Link As from the pop-up menu.
3. Save the file to the **`mcserver/plugins`** folder. If there is no **`plugins`** folder, it's probably because you haven't yet launched CanaryMod. The first time CanaryMod is launched, it creates a couple of configuration files and a **`plugins`** folder, so make sure you've launched CanaryMod at least once before attempting to download ScriptCraft.

If there is already a **`scriptcraft.jar`** file in your **`plugins`** folder (if this isn't the first time you've installed ScriptCraft), make sure to remove it first before saving the new file.

If CanaryMod is running, stop it by issuing the **`stop`** command in the server console.

Launch CanaryMod by double-clicking the launch script you created in Chapter 1. When CanaryMod starts up, you should see an "Enabling scriptcraft" message appear in the server console. The first time it's loaded, the ScriptCraft plugin will unzip a lot of files into a new **`mcserver/scriptcraft`** folder. Your server console output might look something like **FIGURE 2.1**.

```
Enabling scriptcraft v3.0.0-2014-10-14
Directory /home/walter/mcserver/plugins/scriptcraft does not exist.
Initializing /home/walter/mcserver/scriptcraft directory with contents from plugin archive.
Unzipping /home/walter/mcserver/scriptcraft/lib/command.js (NE)
Unzipping /home/walter/mcserver/scriptcraft/lib/console.js (NE)
Unzipping /home/walter/mcserver/scriptcraft/lib/events.js (NE)
```

FIGURE 2.1 Unzipping files

There will be many more entries because many files are bundled with ScriptCraft.

The **`mcserver/plugins`** subfolder is important. It's where all Minecraft server plugins should be stored. This is true not just for **`ScriptCraft.jar`** but for all server plugins. If you save a Server plugin JAR file to any other location except the plugins folder, then it will not be loaded when CanaryMod starts. CanaryMod looks only in the **`plugins`** folder for plugins to load at startup.