The all in one RSPS Guide

By Poanizer

-Disclaimer-

I have not learnt java professionally, I self-taught myself with trial and error. Some of the information in here may be wrong. Use at own risk, I am not to blame if people correct you and you look dumb. This is just my interpretation and knowledge of the java RSPS coding. All terms used may not be the correct terms for proper java coding, yet again this is just how I interpret it.

If in doubt, Google it.

Contents

The most Asked Question 4

What is Revision? 4

Main Revisions 5

317: 5

503: 5

667: 6

718: 6

742: 7

Getting Started with RSPS's 7

What is a source 7

What is a client 8

What java do you need 9

Recommended tools 10

Notepad++ 11

Sublime Text 11

Eclipse 12

What is Compiling? 13

Main files to edit 14

Player.Java 14

Settings.Java 15

Coordinates 17

What is Syntax 18

Java operations and other handy stuff 21

Math Signs: 21

Parentheses 22

If Statements 22

How to use "Or" and "And" 23

Variables........ 24
What is a Variable? 24
How to create a variable 24
How to set a variable by default 26
How to set your Variable in code 26
Calling/Referencing Variables: 27
A Real Example: 27
Functions and Arguments 29
Functions 29
Arguments 30
Imports........ 33
What are Imports?........ 33
What is a Path? 33
Back to Imports........ 34
Java Imports........ 35
Examples of Java Features: 36
Common RSPS Errors. 36
How to Read Errors........ 36
Cannot find the symbol........ 37
100 Errors 37
Reached end of file while parsing 38
Exception in thread main 39
Null Pointer Exception 39
How to Port Forward........ 40
Ipconfig 40
Making your IPv4 Static 42
How to setup a Hamachi Server........ 45
Why choose a Hamachi Network? 45
How to set up a Hamachi Network 45
How to Join a Hamachi Network........ 49
Virtual Private Servers - VPS 50
What you need to know about VPS's........ 50
Minimum VPS Requirements: 51
What is the Best VPS Company........ 51

VPS Land: 52
BurstNet: 52
TrentaHost: 53
Allowing a port through Firewall 54
For a Local Setup or Hamachi Server. 54
For a VPS running Windows 2008 Server 55
Allowing the port through 57
Dangers of RSPS's 61
Exe Commands 61
Visible Ip Address 62
Java Drive By 63
Thanks/Credits 63
End Review 64

The most Asked Question

How do I create a RSPS?

Everyone wants to make a Runescape Private Server, but they either run into problems or cannot find any guides/tutorials on it. This guide is hopefully enough of a starter to help you get into the RSPS scene and start coding and developing your own Server.

Firstly though I can only give advice on how to make a 667+ RSPS as I have never created any other RSPS besides 667. 317 or 503's will be coded slightly differently, so while some of this guide may work. Not all of it will.

But anyway besides that, let's get started.

What is Revision?

Note: There may be some incorrect details in here it's my interpretation and I personally prefer 667.

"***Revision***" is basically a point in time for Runescape. Each time new updates come out generally there will be a new revision. The revision will have the style and content of Runescape of that time.

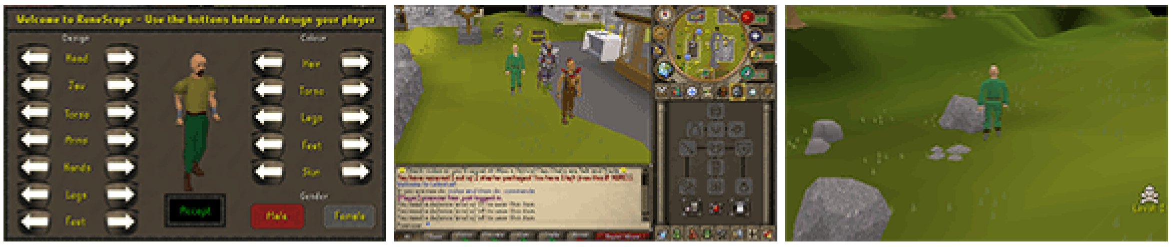
When there are multiple revisions on a server it means that the server is running the base number, with content up to the last number.

Example: 667-718 The base Code is 667 but it has content from the 711/714/718 points of Runescape.

Each main revision is coded differently and uses different files. That is why if you follow my 667 tutorial on your 718 server it may not work. This is also very important when looking for Client and Webclient creation tutorials since each revision has slightly different client files as well.

Main Revisions

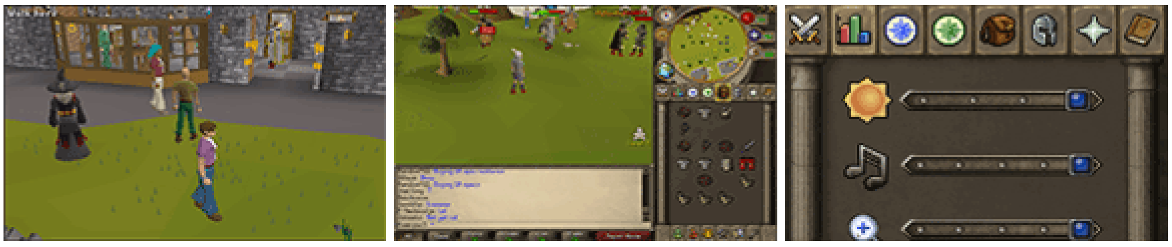
317:


Welcome to RuneScape - Use the buttons below to design your player

Images from: Celestial

- 317 are based around 2006 Runescape.
- The graphics are the original blocky look
- It's said to be the easiest to code with more basic syntax.
- Most Current 317 Servers have newer items and graphics in them.
- The normal main file for Jar'ing a 317 client I believe is ***Client.java***

503:

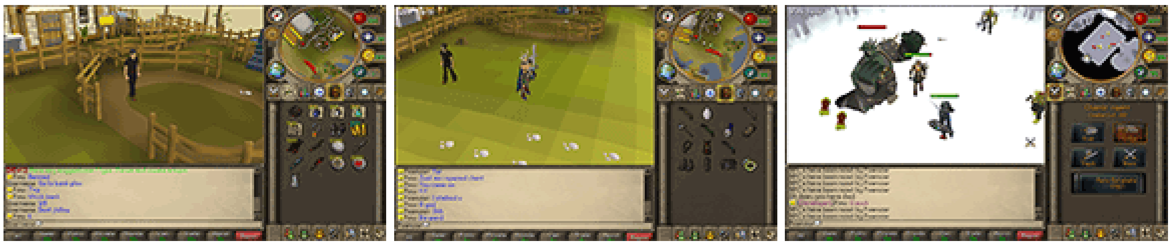


Images from: Devious PK

- 503 are after the graphics have been updated from the blocky ones.

- 503's are said to be harder than 317
- Most 317 Servers on Rune Locus seem like they are 503.
- Honestly I don't know much about this area of revision.

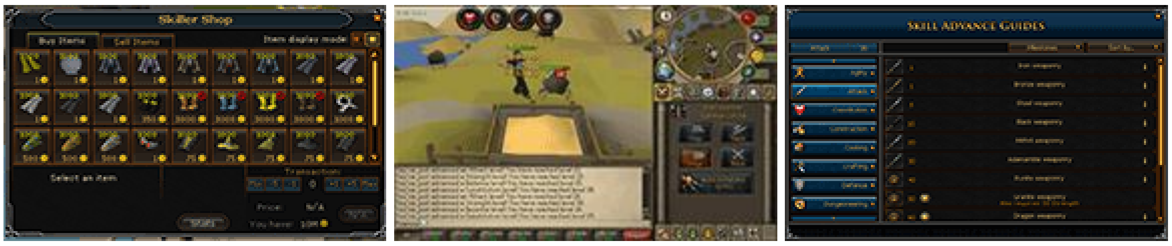
667:



Images from: Extinction v2

- 667 are after the graphics got upgraded.
- 667 is one of the most difficult to code and difficult doesn't vary much in the next few revisions.
- The most popular 667 source released was Zenith which was filled with glitches and bugs.
- There were a lot of 667 sources, but most are being replaced by 718's due to being smoother and less buggy.
- The normal main file for Jar'ing a 667 client is Loader.java

718:



Images from: MBScape Reloaded

- 718 are around the time when the interfaces were updated.
- 718 source releases were a big step from the 667's due to their low bugs/glitches.
- The Squeal of fortune and money pouch are popular updates within 718 sources.

<u>Note</u>
I may be bias when talking about 718+, since I really don't like them. While it's logical to replace buggy servers with these newer ones, I just hold a grudge against them. It is also probably because it was around this point in time when I quit RS.

742:

Images from: PerfectionX

- As far as I'm aware 742 base sources are not released, but 718's are loading more 742 content.
- 718-742's are good due to having EOC items in the original combat style.
- I don't have much information on 742's due to me preferring 667 and they are fairly new. This information could change over time. It was written on 27/10/2013.

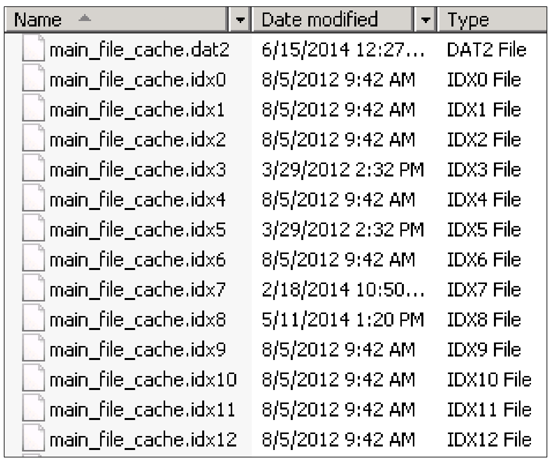
Note:

I think recently they have released source code for EoC servers that have the adrenaline bar and everything else. This was just ripped and slightly edited from my old website.

Getting Started with RSPS's

What is a source

A source is the files of your Server you will be running. The source is the bulk of the server and runs off a "***Cache***".



Name	Date modified	Type
main_file_cache.dat2	6/15/2014 12:27...	DAT2 File
main_file_cache.idx0	8/5/2012 9:42 AM	IDX0 File
main_file_cache.idx1	8/5/2012 9:42 AM	IDX1 File
main_file_cache.idx2	8/5/2012 9:42 AM	IDX2 File
main_file_cache.idx3	3/29/2012 2:32 PM	IDX3 File
main_file_cache.idx4	8/5/2012 9:42 AM	IDX4 File
main_file_cache.idx5	3/29/2012 2:32 PM	IDX5 File
main_file_cache.idx6	8/5/2012 9:42 AM	IDX6 File
main_file_cache.idx7	2/18/2014 10:50...	IDX7 File
main_file_cache.idx8	5/11/2014 1:20 PM	IDX8 File
main_file_cache.idx9	8/5/2012 9:42 AM	IDX9 File
main_file_cache.idx10	8/5/2012 9:42 AM	IDX10 File
main_file_cache.idx11	8/5/2012 9:42 AM	IDX11 File
main_file_cache.idx12	8/5/2012 9:42 AM	IDX12 File

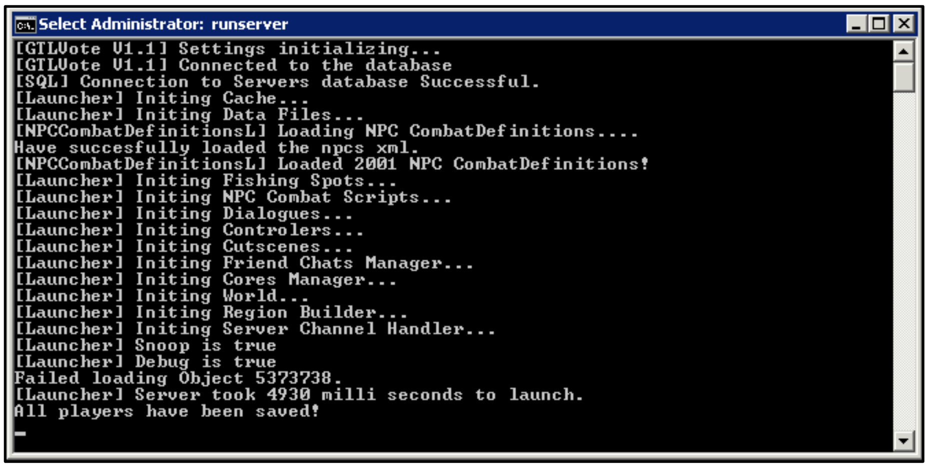
Above: A picture of some of the cache files.

The cache holds most of the actual information and data about your server. It has the Models, Sounds, Definitions, Images and a lot more inside it. The "Default" cache file size is around 500mb. But depending on how much models and "Customs" you have added inside it, it may vary in size.

The other files use the data inside the cache along with coding to put everything together and run a server. Obviously it is a lot more technical than that, but that is just the general explanation.

To start the server you will need to open "***Run.cmd***" which starts the command prompt window that will handle all your server messages and information. Each time you re-open ***Run.*****cmd** It will reload all the files in the source folder in case any changes were made.

Below: Run.cmd initiating all the main files.



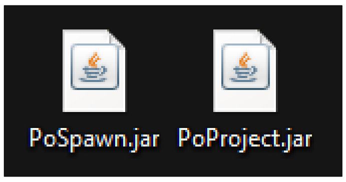
```
[GTLVote V1.1] Settings initializing...
[GTLVote V1.1] Connected to the database
[SQL] Connection to Servers database Successful.
[Launcher] Initing Cache...
[Launcher] Initing Data Files...
[NPCCombatDefinitionsL] Loading NPC CombatDefinitions....
Have succesfully loaded the npcs xml.
[NPCCombatDefinitionsL] Loaded 2001 NPC CombatDefinitions!
[Launcher] Initing Fishing Spots...
[Launcher] Initing NPC Combat Scripts...
[Launcher] Initing Dialogues...
[Launcher] Initing Controlers...
[Launcher] Initing Cutscenes...
[Launcher] Initing Friend Chats Manager...
[Launcher] Initing Cores Manager...
[Launcher] Initing World...
[Launcher] Initing Region Builder...
[Launcher] Initing Server Channel Handler...
[Launcher] Snoop is true
[Launcher] Debug is true
Failed loading Object 5373738.
[Launcher] Server took 4930 milli seconds to launch.
All players have been saved!
_
```

If you get an error when trying to open ***Run.cmd*** check the section about "***Paths***" or "***Common RSPS Errors***" and see if that helps.

What is a client

The client is what you use to connect to the server that is running of the "***Source***" files. You load the client and play through it. The client sends the information back to the server as things called "***Packets***".

Raw Clients are a Java executable file which has an extension of "***.jar***". If you know how, you can make it so your client boots up with an "***.exe***" launcher normally these are more customised. Another option is you can embed the ***.jar*** so it launches in a webpage. This is referred to as a "***Web-Client***".



Above: Examples of 2 "Raw" Clients.

What java do you need

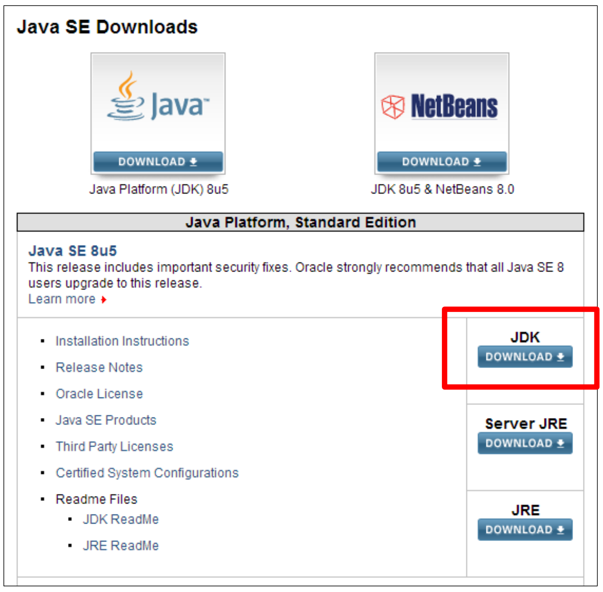
To setup and run a RSPS you will need to download a version of the ***Java Development Kit*** which is also referred to as ***JDK***.

My knowledge on the technical reason for the JDK is quite small but all I know is that the JDK is used for compiling.

You can find the latest version of the JDK at:

http://www.oracle.com/technetwork/java/javase/downloads/index.html?ssSourceSiteId=otnjp

When your there make sure you click "***JDK***" and select the correct version that matches what operating system your running on.



Java SE Downloads

Java Platform (JDK) 8u5

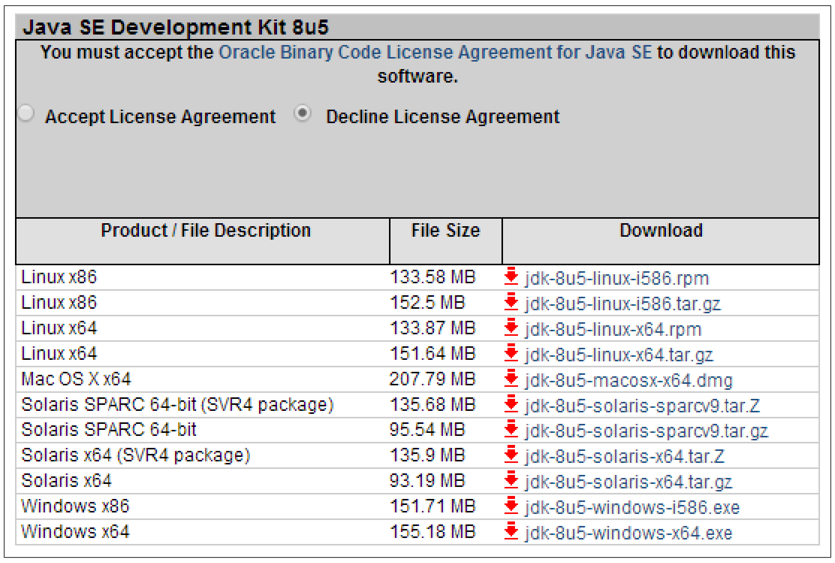
JDK 8u5 & NetBeans 8.0

Java Platform, Standard Edition

Java SE 8u5
This release includes important security fixes. Oracle strongly recommends that all Java SE 8 users upgrade to this release.
Learn more

- Installation Instructions
- Release Notes
- Oracle License
- Java SE Products
- Third Party Licenses
- Certified System Configurations
- Readme Files
  - JDK ReadMe
  - JRE ReadMe

Make sure you match the "***Product/File Description***" with your operation system.



Java SE Development Kit 8u5

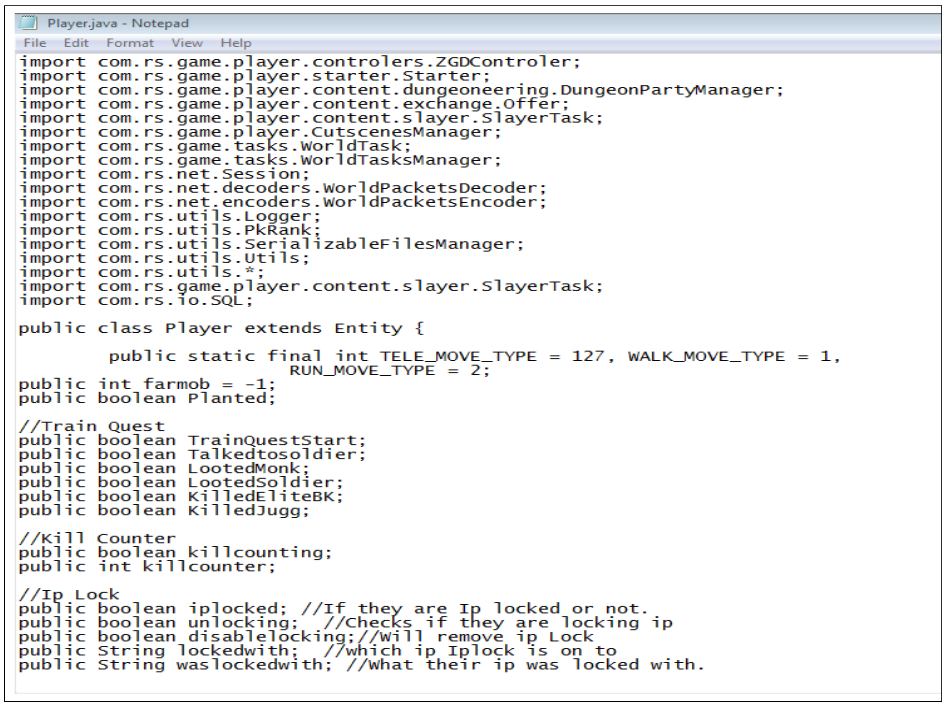
You must accept the Oracle Binary Code License Agreement for Java SE to download this software.

Accept License Agreement [✓ Decline License Agreement]

Product / File Description	File Size	Download
Linux x86	133.58 MB	jdk-8u5-linux-i586.rpm
Linux x86	152.5 MB	jdk-8u5-linux-i586.tar.gz
Linux x64	133.87 MB	jdk-8u5-linux-x64.rpm
Linux x64	151.64 MB	jdk-8u5-linux-x64.tar.gz
Mac OS X x64	207.79 MB	jdk-8u5-macosx-x64.dmg
Solaris SPARC 64-bit (SVR4 package)	135.68 MB	jdk-8u5-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	95.54 MB	jdk-8u5-solaris-sparcv9.tar.gz
Solaris x64 (SVR4 package)	135.9 MB	jdk-8u5-solaris-x64.tar.Z
Solaris x64	93.19 MB	jdk-8u5-solaris-x64.tar.gz
Windows x86	151.71 MB	jdk-8u5-windows-i586.exe
Windows x64	155.18 MB	jdk-8u5-windows-x64.exe

Recommended tools

When you start coding a RSPS you need to find the correct software to edit the code in. ***Notepad*** or ***Microsoft Word*** does not work well as they aren't designed for coding in. It is best to find software that will colour the main features of the language and allow you to indent your code easily. Indenting code is another way of saying: add spaces before a line so it sticks out more.

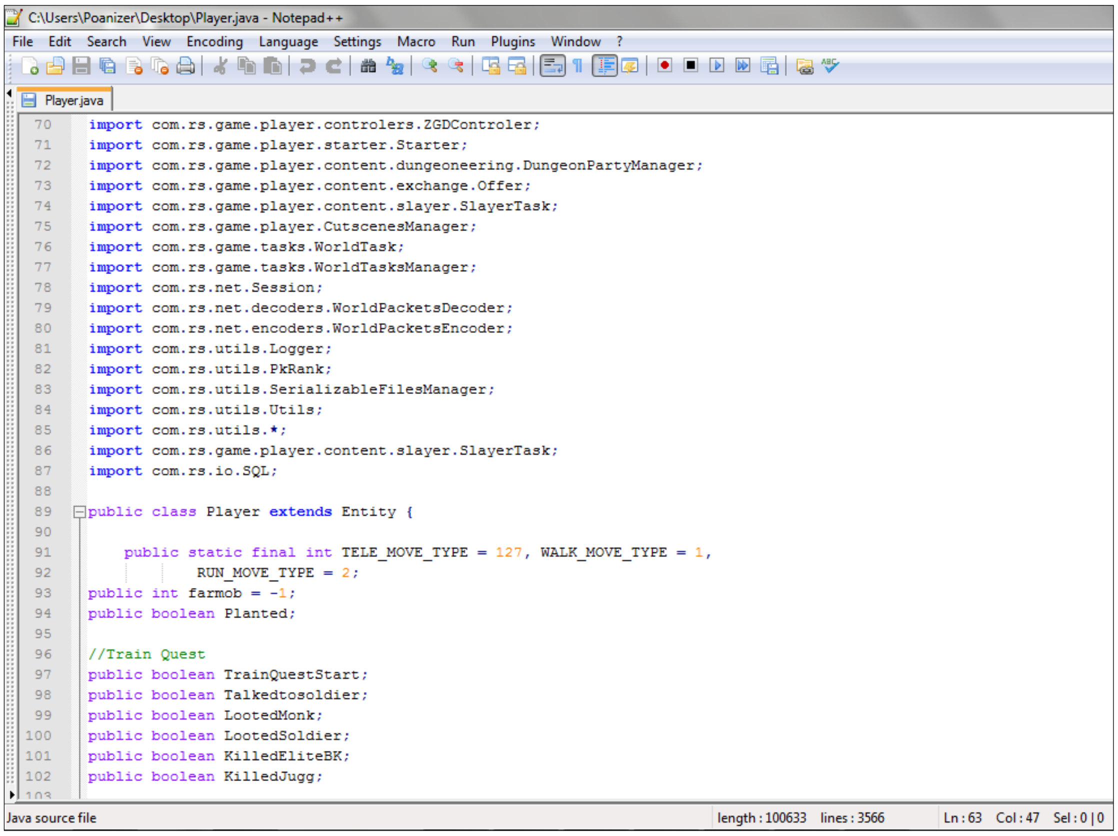


```
import com.rs.game.player.controlers.ZGDControler;
import com.rs.game.player.starter.Starter;
import com.rs.game.player.content.dungeoneering.DungeonPartyManager;
import com.rs.game.player.content.exchange.Offer;
import com.rs.game.player.content.slayer.SlayerTask;
import com.rs.game.player.CutscenesManager;
import com.rs.game.tasks.WorldTask;
import com.rs.game.tasks.WorldTasksManager;
import com.rs.net.Session;
import com.rs.net.decoders.WorldPacketsDecoder;
import com.rs.net.encoders.WorldPacketsEncoder;
import com.rs.utils.Logger;
import com.rs.utils.PkRank;
import com.rs.utils.SerializableFilesManager;
import com.rs.utils.Utils;
import com.rs.utils.*;
import com.rs.game.player.content.slayer.SlayerTask;
import com.rs.io.SQL;

public class Player extends Entity {

        public static final int TELE_MOVE_TYPE = 127, WALK_MOVE_TYPE = 1,
                RUN_MOVE_TYPE = 2;
public int farmob = -1;
public boolean Planted;

//Train Quest
public boolean TrainQuestStart;
public boolean Talkedtosoldier;
public boolean LootedMonk;
public boolean LootedSoldier;
public boolean KilledEliteBK;
public boolean KilledJugg;

//Kill Counter
public boolean killcounting;
public int killcounter;

//Ip Lock
public boolean iplocked; //If they are Ip locked or not.
public boolean unlocking;  //Checks if they are locking ip
public boolean disablelocking;//Will remove ip Lock
public String lockedwith;  //which ip Iplock is on to
public String waslockedwith; //What their ip was locked with.
```

Above: Player.java opened in Windows Notepad.

Notepad++

My recommendation is to use ***Notepad++*** this is software designed like the original Notepad in windows but with extra features developed for people who are editing code. Notepad++ colour codes main features in java and comes with some basic helpful features such as Bulk Replacing, Spellcheck and numbered lines.


C:\Users\Poanizer\Desktop\Player.java - Notepad++
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
Player.java
Java source file
length : 100633 lines : 3566
Ln : 63 Col : 47 Sel : 0 | 0

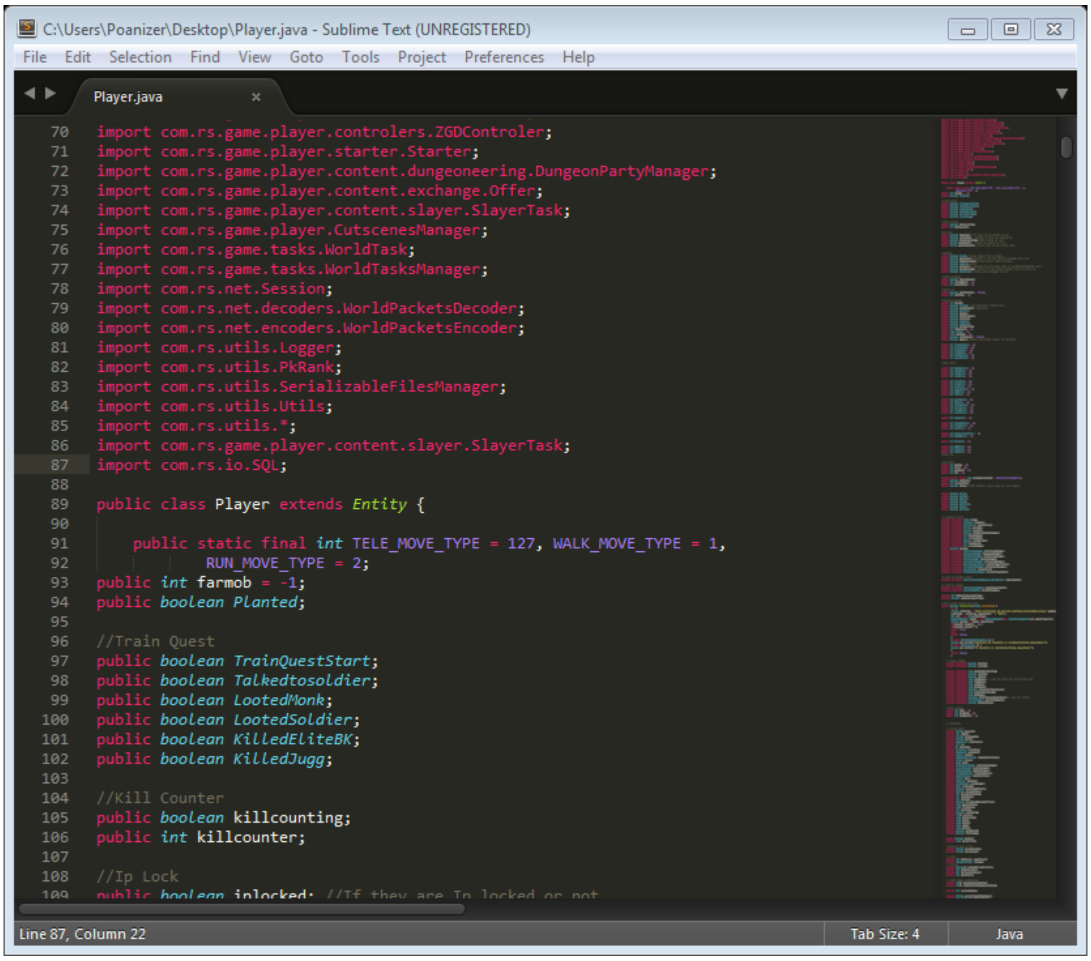
```java
import com.rs.game.player.controlers.ZGDControler;
import com.rs.game.player.starter.Starter;
import com.rs.game.player.content.dungeoneering.DungeonPartyManager;
import com.rs.game.player.content.exchange.Offer;
import com.rs.game.player.content.slayer.SlayerTask;
import com.rs.game.player.CutscenesManager;
import com.rs.game.tasks.WorldTask;
import com.rs.game.tasks.WorldTasksManager;
import com.rs.net.Session;
import com.rs.net.decoders.WorldPacketsDecoder;
import com.rs.net.encoders.WorldPacketsEncoder;
import com.rs.utils.Logger;
import com.rs.utils.PkRank;
import com.rs.utils.SerializableFilesManager;
import com.rs.utils.Utils;
import com.rs.utils.*;
import com.rs.game.player.content.slayer.SlayerTask;
import com.rs.io.SQL;

public class Player extends Entity {

    public static final int TELE_MOVE_TYPE = 127, WALK_MOVE_TYPE = 1,
            RUN_MOVE_TYPE = 2;
public int farmob = -1;
public boolean Planted;

//Train Quest
public boolean TrainQuestStart;
public boolean Talkedtosoldier;
public boolean LootedMonk;
public boolean LootedSoldier;
public boolean KilledEliteBK;
public boolean KilledJugg;
```

Above: Player.java opened in Notepad++.

Sublime Text

Another alternative which is normally used in web design is software called ***Sublime Text***. This software however isn't free and you will need to buy a license or use the trial.

This software is similar to ***Notepad++*** However the main colour schemes are a lot darker along with some features such as multiple point typing. This means you are able to select multiple points on any line and type/delete on all of them at once. It also has most of the features Notepad++ has.


C:\Users\Poanizer\Desktop\Player.java - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
Player.java
Line 87, Column 22
Tab Size: 4
Java

```java
import com.rs.game.player.controlers.ZGDControler;
import com.rs.game.player.starter.Starter;
import com.rs.game.player.content.dungeoneering.DungeonPartyManager;
import com.rs.game.player.content.exchange.Offer;
import com.rs.game.player.content.slayer.SlayerTask;
import com.rs.game.player.CutscenesManager;
import com.rs.game.tasks.WorldTask;
import com.rs.game.tasks.WorldTasksManager;
import com.rs.net.Session;
import com.rs.net.decoders.WorldPacketsDecoder;
import com.rs.net.encoders.WorldPacketsEncoder;
import com.rs.utils.Logger;
import com.rs.utils.PkRank;
import com.rs.utils.SerializableFilesManager;
import com.rs.utils.Utils;
import com.rs.utils.*;
import com.rs.game.player.content.slayer.SlayerTask;
import com.rs.io.SQL;

public class Player extends Entity {

    public static final int TELE_MOVE_TYPE = 127, WALK_MOVE_TYPE = 1,
            RUN_MOVE_TYPE = 2;
public int farmob = -1;
public boolean Planted;

//Train Quest
public boolean TrainQuestStart;
public boolean Talkedtosoldier;
public boolean LootedMonk;
public boolean LootedSoldier;
public boolean KilledEliteBK;
public boolean KilledJugg;

//Kill Counter
public boolean killcounting;
public int killcounter;

//Ip Lock
public boolean iplocked; //If they are Ip locked or not
```

Above: Player.java opened in Sublime Text

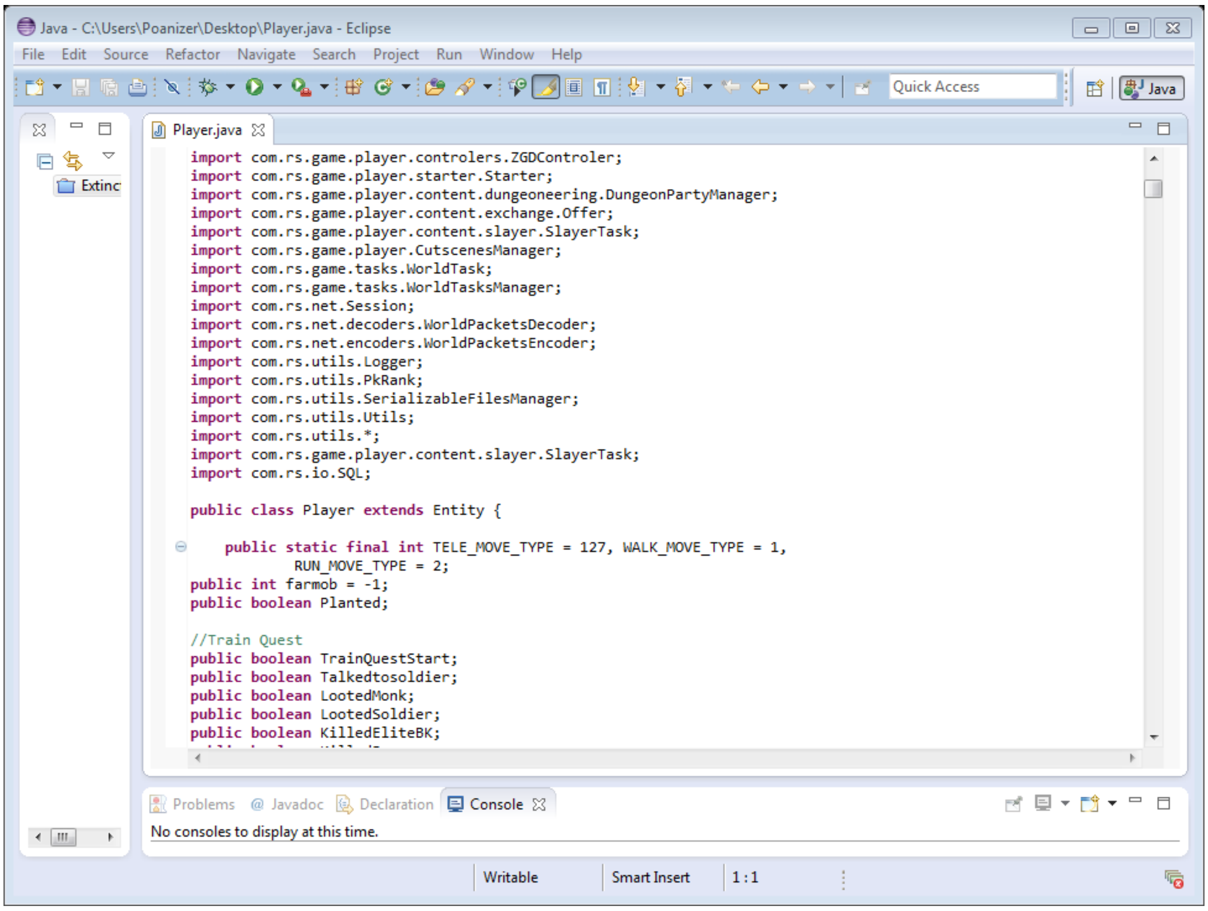
Eclipse

The final alternative that I know about is a programme called eclipse. Personally I hate eclipse and don't use it, but a lot of people recommend it and it is used by most "***Professional***" Coders.

It auto compiles everything instantly so there is no need to run a ***Compiler***. It will also help explain any errors you get. There is also a navigation pane on the side where you can browse the rest of the files in your source folder.

This layout however can all be customised to suit your liking. For beginners the layout and setup may be confusing, since it uses a more professional developer setup, but it will probably be fine after you learn how to navigate and use it.

I didn't bother to learn how to use it so I just loaded the Player.java file into eclipse instead of using my whole source as a "***Project***".


Java - C:\Users\Poanizer\Desktop\Player.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Java
Extinc
Player.java
Problems
Javadoc
Declaration
Console
Writable
Smart Insert
1 : 1

```java
import com.rs.game.player.controlers.ZGDControler;
import com.rs.game.player.starter.Starter;
import com.rs.game.player.content.dungeoneering.DungeonPartyManager;
import com.rs.game.player.content.exchange.Offer;
import com.rs.game.player.content.slayer.SlayerTask;
import com.rs.game.player.CutscenesManager;
import com.rs.game.tasks.WorldTask;
import com.rs.game.tasks.WorldTasksManager;
import com.rs.net.Session;
import com.rs.net.decoders.WorldPacketsDecoder;
import com.rs.net.encoders.WorldPacketsEncoder;
import com.rs.utils.Logger;
import com.rs.utils.PkRank;
import com.rs.utils.SerializableFilesManager;
import com.rs.utils.Utils;
import com.rs.utils.*;
import com.rs.game.player.content.slayer.SlayerTask;
import com.rs.io.SQL;

public class Player extends Entity {

    public static final int TELE_MOVE_TYPE = 127, WALK_MOVE_TYPE = 1,
            RUN_MOVE_TYPE = 2;
public int farmob = -1;
public boolean Planted;

//Train Quest
public boolean TrainQuestStart;
public boolean Talkedtosoldier;
public boolean LootedMonk;
public boolean LootedSoldier;
public boolean KilledEliteBK;
```

No consoles to display at this time.

Above: Player.java opened in Eclipse

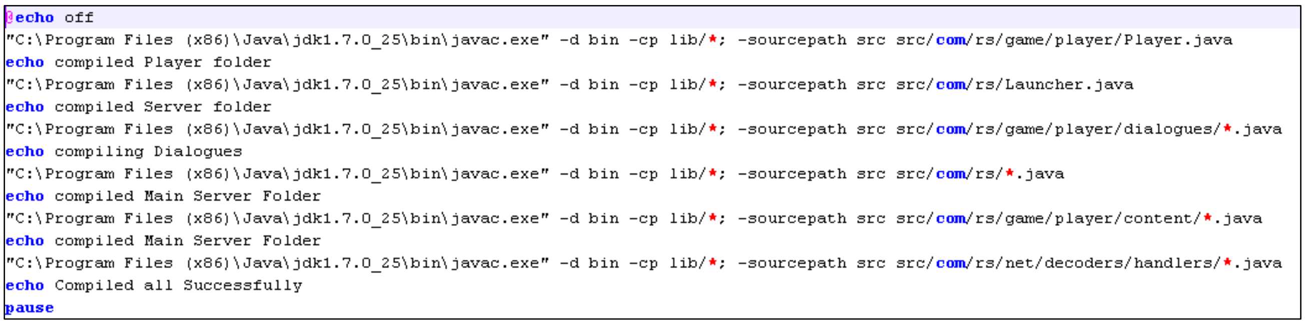
What is Compiling?

Compiling basically makes the edits you have made in the source come into effect when you restart your server. You may notice you edit a file and yet when you restart the server nothing is different.

When you open your source folder there will be 2 folders that carry all the files. There will be a "***src***" and a "***bin***" folder. The ***src*** folder is where all the files are in ***.java*** format. These files are the ones you edit to make the changes in. The ***bin*** folder has the same files as the ***src*** but in this one you will find all the files are in "***.Class***" file type.

You should also see a file called "***Compiler.bat***" in your source folder. This is the default compiler we will be talking about in this section. If you use eclipse it should compile everything for you. You can also make custom compilers that compile different folders.

When you open the compiler you should see the following….


```
@echo off
"C:\Program Files (x86)\Java\jdk1.7.0_25\bin\javac.exe" -d bin -cp lib/*; -sourcepath src src/com/rs/game/player/Player.java
echo compiled Player folder
"C:\Program Files (x86)\Java\jdk1.7.0_25\bin\javac.exe" -d bin -cp lib/*; -sourcepath src src/com/rs/Launcher.java
echo compiled Server folder
"C:\Program Files (x86)\Java\jdk1.7.0_25\bin\javac.exe" -d bin -cp lib/*; -sourcepath src src/com/rs/game/player/dialogues/*.java
echo compiling Dialogues
"C:\Program Files (x86)\Java\jdk1.7.0_25\bin\javac.exe" -d bin -cp lib/*; -sourcepath src src/com/rs/*.java
echo compiled Main Server Folder
"C:\Program Files (x86)\Java\jdk1.7.0_25\bin\javac.exe" -d bin -cp lib/*; -sourcepath src src/com/rs/game/player/content/*.java
echo compiled Main Server Folder
"C:\Program Files (x86)\Java\jdk1.7.0_25\bin\javac.exe" -d bin -cp lib/*; -sourcepath src src/com/rs/net/decoders/handlers/*.java
echo Compiled all Successfully
pause
```

This is what the general 667 compiler looks like. The first line in the code is the path to javac.exe, the Java compiler in your ***JDK***. After the "***-sourcepath src***" is the path to the folder/files you will be compiling. These are all files inside the ***src*** folder.

The star ***** in the path is used as a wildcard. This means that anything with a ***.java*** extension in that folder will be compiled. If it doesn't have the wildcard then it will use the name of the file instead.

Example: *rs/Launcher.java* would compiler just the Launcher folder.

rs/.java* would compile all the files that have a .java extension in the "***rs***" folder.

Main files to edit

When starting your own 667 Server the main files you will need to edit in the Source are "***Player***" and "***Settings***".

Player.java handles all the rights and features for every player that logs in.

Its location is: src/com/rs/game/player/***Player.java***

Settings.java handles all the settings for the server, such as "Home" and Respawn locations along with the Server Name.

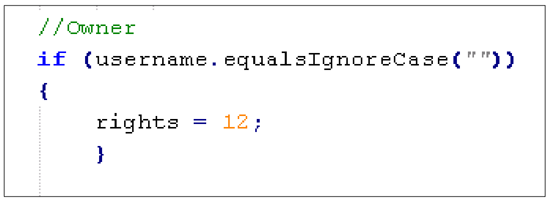
Its location is: src/com/rs/***Settings.java***

Player.Java

To start I will show you what you need to change in **Player.java**. This is not required as generally you can edit these settings ingame via commands but it is best you edit/remove them in Player.java first.

Step one:

When you have player.java open, Push ***Ctrl + F*** or find the “Search” feature of your text editor. Look for “***Rights***”. Rights are another word for “***Rank***”. Keep looking until you find code that look like the below image.



```
//Owner
if (username.equalsIgnoreCase(""))
{
    rights = 12;
    }
```

Depending on which source you’re using the number that is used for each rank will be different.

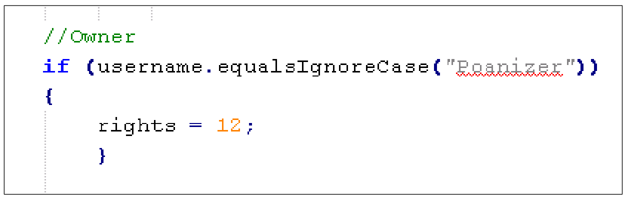
Example:

The Poanizer Project’s Owner Right is: **12**

Zenith’s Owner Right is: **7**

Step Two:

Inside the two quote marks **“ ”** type in your desired Username for the account you will use on your Server. For this example we will be using the username “***Poanizer***”. Remember to be original and make your own name. Nobody likes a copy.



```
//Owner
if (username.equalsIgnoreCase("Poanizer"))
{
    rights = 12;
    }
```

That is all you need to do to make yourself have Owner Rank ingame. Whatever number is after “**rights =**” will be your rank. Make sure to check any other pieces of code that look like that one to make sure no other usernames are set to have ranks.

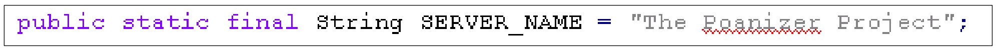
And do remember to read about “***Compiling***” above, or else these changes won’t save when you restart your server.

Settings.Java

Now that you have changed your ranks it would be time to change your Server’s Name and the Home along with the respawn Location.

Step one:

When you have ***Settings.java*** opened, you should see at the top of the page, depending on your source, something that says "***SERVER NAME***".

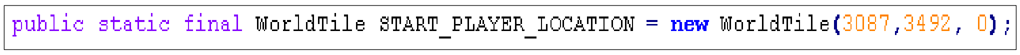


```
public static final String SERVER_NAME = "The Poanizer Project";
```

Edit the Text in between the two Quotes **" "** again to whatever your server name will be. In this example the server name is "***The Poanizer Project***".

Step Two:

After that is finished and you have entered your desired Server Name, push ***Ctrl + F*** or use the search feature and look for "***Location***". Keep looking until you find the below piece of code.



```
public static final WorldTile START_PLAYER_LOCATION = new WorldTile(3087,3492, 0);
```

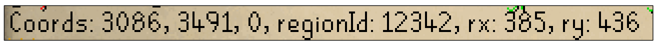
When you have found that login to your server and find the tile you wish players to start on. Then use the command "***Coords***" ingame to get the coordinates for that specific tile. If you are unaware about what coordinates are please read the explanation below.

Once you have the coordinates for that tile, change the first number in the brackets **( )** to the first number of your tile's coordinates. Then repeat the same with the others.

The order is: WorldTile **(X**, **Y**, **Z)**

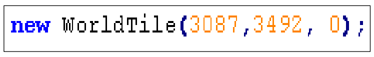
Example:

If this was the result from doing the command ***::coords*** ingame…



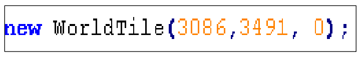
```
Coords: 3086, 3491, 0, regionId: 12342, rx: 385, ry: 436
```

You would Replace:



```
new WorldTile(3087,3492, 0);
```

With:

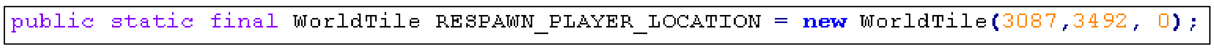


```
new WorldTile(3086,3491, 0);
```

Which has the coordinates you got from the command.

Step Three:

After you have set the start location, scroll down a few lines or push ***Ctrl + F*** or use the search feature again to look for "***Respawn***". Keep looking until you find the below piece of code.



```
public static final WorldTile RESPAWN_PLAYER_LOCATION = new WorldTile(3087,3492, 0);
```

Use the same method that was explained above to set the Location that the players will respawn to when they die.

After you have completed that you would have fixed all the basic changes for new servers.

Coordinates

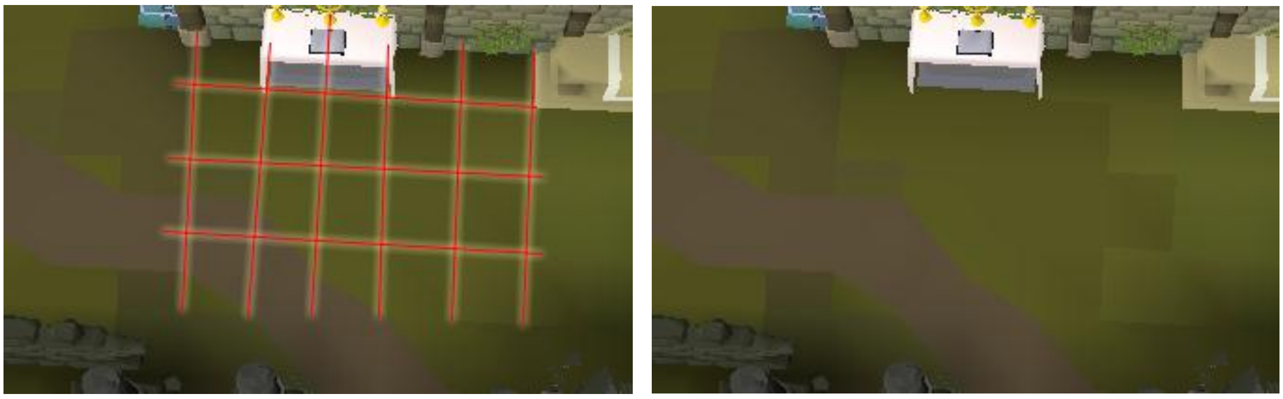
The numbers inside the Brackets **( )** are coordinates for a Tile ingame. Runescape is sectioned up by a massive grid. Each step you walk is equivalent to one tile. And each tile has its own coordinates. There is an ***X, Y, Z*** coordinate.

The **X** Coordinate is the tile number Horizontal to the map.

The **Y** is the tile number Vertical to the map.

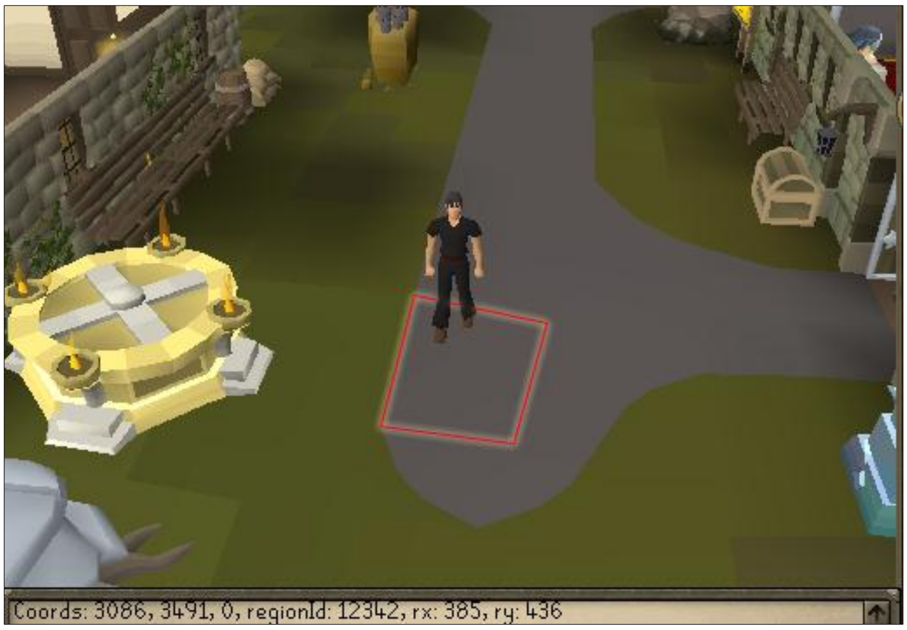
The **Z** is the height level on the map.

Example: Below is an image which shows how the floor looks along with how it theoretically looks as a grid. Each square in the grid would be a tile. You can also see some of the tiles with their different colours on the image to the right.



If you go ingame on your server and have owner rank, you can use the command "**Coords**". The name of this command may vary depending on the source. What this will do is get your current standing location's coordinates.

Example: In the Chatbox at the bottom it shows the current **X, Y, Z** Coordinate along with extra information for the tile I am standing on. (I drew a rough square around the tile)



```
Coords: 3086, 3491, 0, regionId: 12342, rx: 385, ry: 436
```

Other examples of where coordinates are used are for spawning NPC's and Objects.

__**NPCSpawning.java**__:

Objects: the below image has an example of a spawned object and underneath what each id means.

```
World.spawnObject(new WorldObject(1, 10, 3, 2090, 5795, 0), true);
World.spawnObject(new WorldObject(object id, type, rotation, x, y, z), true);
```

NPC's: the below image has an example of a spawned NPC and underneath what each id means.

```
World.spawnNPC(13295, new WorldTile(2326, 3801, 0), 0, true, true);
World.spawnNPC(NPC id, new WorldTile(x, y, z), 0, true, true);
```

What is Syntax

Syntax is like the layout. The layout for a basic piece of java code goes.

if statement (**Variable** operation **Variable**)

open bracket **{**

```
        "code that the statement implements";
      } closed bracket
```

End result:

```
If (Variable ==  Variable) {
"code";
}
```

Another Example:

```
if (money  >=  50) {
        Say: "I'm Rich"
}
```

Understanding so far?

Each open bracket, this: **{**

Must have a closing bracket, this: **}**

If the **closing bracket** is not there then it will use the next closing bracket. But that will break the next bit of code since its closing bracket just disappeared. This broken code will then use the next closing bracket, breaking every piece of code until it reaches the end of the file.

This type of break will generally result in a "100 Errors" message, because each piece of code after the one missing a bracket will break.

Below is a conversation that took place on Skype. I try to explain how Brackets { } work. To make it easier for me I copied and pasted the conversation and Taivo, the person I was explaining it too has let me use it.

You may think it is completely dumb and skip it. I don't mind either way.

Banana 1 has a handle thingy (You know how bananas have those handle things?) and it has a black bottom cap.

Banana 1 is above banana 2

Banana 1 is missing his bottom cap, so he will take banana 2's bottom cap instead

now banana 2 doesn't have a bottom so he tries to take the next available bottom. But since there is no bottom cap then he gets all confused and won't work properly.

***taivo**: you mean if I make a statement I need to close it with brackets?*

Yea, any piece of code, except for some exceptions use brackets

- Open **{**
- Close **}**

Inside those brackets is the code that the statement runs.

So if I wrote this code :

if (**Poanizer** == **Male**) **{**

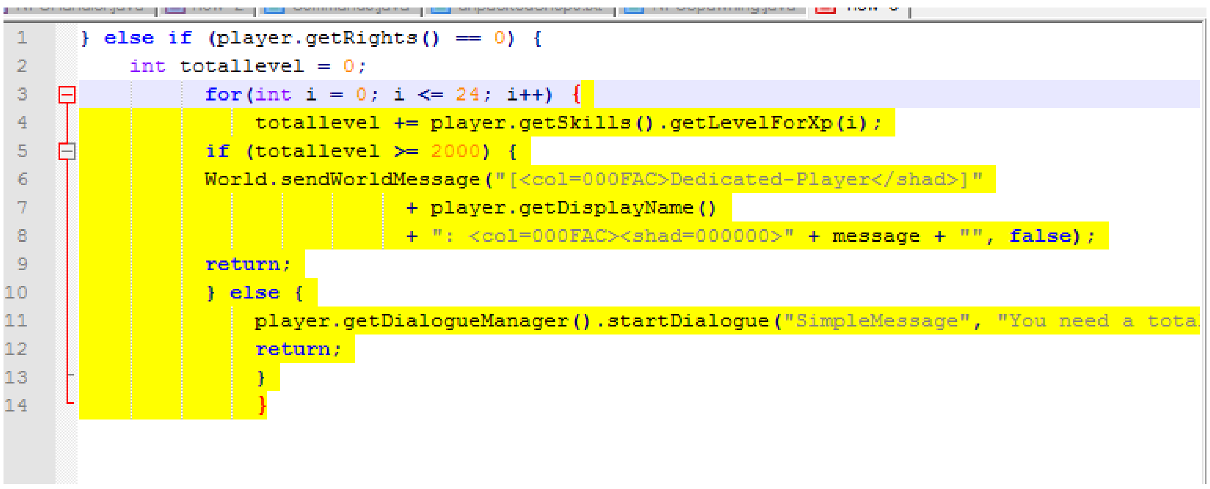
Walk 50 meters.

}

That would make me walk 50 meters since I am a male. Or you could say that the statement is true, as in it works. Poanizer is a male. What it does is it checks IF(Poanizer "is" male).

This screenshot may help.

If you still don't understand I have highlighted from the open brackets to the closing brackets along with all the code inside.



```
} else if (player.getRights() == 0) {
    int totallevel = 0;
        for(int i = 0; i <= 24; i++) {
            totallevel += player.getSkills().getLevelForXp(i);
        if (totallevel >= 2000) {
        World.sendWorldMessage("[<col=000FAC>Dedicated-Player</shad>]"
                        + player.getDisplayName()
                        + ": <col=000FAC><shad=000000>" + message + "", false);
        return;
        } else {
            player.getDialogueManager().startDialogue("SimpleMessage", "You need a tota
            return;
            }
            }
```

Java operations and other handy stuff

Some of the information in here isn't exactly for java but may be handy if you are making a SQL Database, such as an AutoVote System or a Webclient/Website.

Below is basic "comment" syntax. Comments are handy for adding a note besides pieces of code or a way of adding an explanation of what something does. Comments are not read by the server and will generally be coloured a certain colour if your using a code editor. **Example:** Notepad++.

HTML comment: <!-- --> Used to comment in Html code. A ".Html" file.

Multiple Comment: /* */ This is used in Java or PhP to comment out multiple lines of code.

Single Comment: // This will comment out everything on the line after these to slashes. Php/Java.

Math Signs:

Addition: +

Division: /

Multiplication: *

Less than: <

Greater than: >

Less than or equal to: >=

Greater than or equal to: >=

Equals to : ==

Set to: =

If there is only one equals sign it will make it set the value rather than equalling it to something

Example: 1 = 2: 1 will be set as 2 **Example 2:** 1 == 2: will check if 1 is the same as 2.

Variation Combinations: ++ means add 1. - - means minus 1. And normally if you're adding/subtracting values you would add an equals sign to the end, **Example:** += or -=

The exclamation mark, **!** , means “not” or “Isn’t” , so if you see: **If(!**Player.donator)

It will mean: if player **Isnt** donator. This won’t work in every situation though.

Parentheses

Parentheses are the proper name for the smooth brackets **( )**

Just like the other brackets **{ }** Any open parentheses needs a to be closed.

Open: (

Close:)

Parentheses are normally used in the If statements or any time you need to group things together.

There isn’t much more to say, just follow the previous information on brackets **{ }** and you will be fine. There can be multiple uses of parentheses in one statement. If this is the case always make sure there are no unclosed parentheses.

Example: sendGameMessage**(** “your total xp is “ + **(**player.getxp + **(**2 + 4 * 5 **))**;

Each open bracket above , **(** , has a closing bracket, **)**

If Statements

If statements have their own layout. The first part will be the expression whether it is “If” “else if” or “else”. The order is stated below.

For every statement you start by using **if.**

if: This will check if what you put in the parentheses is true. If you want it to do something else when the first statement isn’t true then….

You would use **else if** after it.

else if: it will check this condition is true. Else if means the same as if, just if you already have an if statement you want to add to, you need to use else if. You can use this multiple times.

Then if you want it to do something even if none of the previous statements are true, use….

else: Normally people use else to show the user they done something wrong. You may use "***else***" straight after an **if** statement, but "***else***" can only be used once.

Example:

if (player has the requirements){

"You have the requirements for this"

}

else{

"You don't have the requirements"

}

How to use "Or" and "And"

If you want to make an "***or***" use 2 lines || . These are found generally above the "enter" key. Or by pushing shift while typing: \ .

Normal \\\\\\

Shift+\ |||||

If you want to use "***and"*** then you use 2 and symbols && , which are generally found using shift and 7.

The main thing to remember when using "***or"*** or "***and"*** is that each thing your adding on, needs to be in its own set of parentheses.

Example: (player.getrights() == 1**)** **||** **(**player.getrights() ==4 **)**

Not: (player.getrights() == 1 **||** player.getrights() ==4 **)** - this one will give an error.

Variables

Do remember I'm not 100% with java lingo so if you actually know anything about java my explaining may be semi incorrect or confusing.

What is a Variable?

A variable is a Thing that is set to have a certain value. In RSPS there are a few main variables.

1. **int**: an int or integer is basically a number with no decimal, ***Example***: 1, 20 , 49
2. **decimal**: a decimal is a number with a decimal place, ***Example***: 1.0, 20.5, 38.9
3. **long**: A long is a value above 2.4 billion, or the max value of an int (Max cash).
4. **boolean**: A boolean is a value that is either "true" or "false". This is handy for doing requirements or to enable/disable features. ***Example***: hasMaxed, ShowDrops, hasVoted.

5. **String**: A string is text. Normally strings will have quotes around them. ***Example***: "Hello", "This is a sentence", "". If you see a string like this "" it means it's a blank/empty string. These are normally best if you're going to set the string to something.

   To add punctuation to a string use \ before the punctuation quote, this will cancel its affect.

   Example: 'That's a good example';   **Fixed**: 'That\'s a good example';

6. **Entity**: Entities I'm still learning about, so the information after this may be incorrect, however it's my interpretation of them. What i understand them to be is like a living moving thing. Or a complex variable. Some examples of an entity you may see in RSPS Coding is: NPC or Player.

   You can also see some alternative entities which fit into the 2 main entities but aren't that exact one. Examples: Trader, partner, p, target.

   ***** *The alternative "entities" are probably not "entities" but that's what I refer them as.*

How to create a variable

To create a Variable you will need to define whether it is a "***public***" or a "***private***" variable.

Public means that other files can reference it, or call upon it. ***Example***: Player.VotePoints

To create a variable you first need to tell the file whether it’s public or private, if you don’t know why these matter then just use "***Public***". Or research more information about java coding.

After declaring it’s a public variable, which means it can be referenced in other files, you will need to add what type of variable you are creating.

The basic types were listed above: ***int***, ***decimal***, ***long***, ***String***, etc.....

For this example we will make an integer, as they are generally the easiest. You would then add "int" after your public/private declaration.

So far we have: "***public int***”.

After you have what type of variable it is, you need to set a name to your variable. This will be what it is called.

Normally coders use "camel case" which makes everything one word but separates each one by capitalising the first letter of each word but leaving the very first letter small.

Example: camelCase, thisIsCamelCase.

For this example we will call our variable "***testNumber***", add that after the type of variable you have.

So far we have: *“**public int testNumber**”*

To close it off and not set it to anything you will need to add a semi colon **;** at the end, the semi colon generally just means end/stop.

Your final result: *“**public int testNumber;**”*

How to set a variable by default

Like I said before variables are things that have values set to them.

If you are creating a **String** or just want to make sure it’s set to the correct value, you will need to set it to something. I normally set the variable regardless.

To set a variable you need to use the operation **=** this stands for "***set to***". Unlike using two of them side by side as in: == , which means "***equals to***".

So to set a variable by default you will need to find the original piece of code where the variable was created.

Our example looks like this: “***public int testNumber***”;

After the Name of the variable, but before the closing semi colon add an equals sign and then the value you wish to set that variable to.

If we wanted to set ***testNumber*** to **588**, we would make it look like this:

public int testNumber = 588;

Remember to keep the semicolon on the end.

How to set your Variable in code

This is for editing the variable from a different file, to the one it’s created in.

If you want to remotely set the variable or set it in some code you just need to use the same method above but only use the variables name and the file it was created in.

(Refer to the below "calling/referencing" guide)

For this example we will be changing the value of ***testNumber*** which we set to 588, to **231** from a different file.

Example: Player.testNumber = 231;

This is to set a variable in the same file that it was created in.

To do this you just follow the above method, but remove the file name in front of it.

Since it’s in the same file, it will find the variable you have created by searching for the variables name.

You need to make sure you type your variable's name case sensitively so the reference one we are using now to change its value, matches the original. Remember you still need to add a semi colon to the end of it.

Example : testNumber = 231;   **Instead of:** Player.testNumber = 231;

Since we are referencing it in Player.java we do not need to add **Player.Java's** file name, which is "**Player**" to the front of the variable.

(Described in more detail below)

Calling/Referencing Variables:

To use the variables that have been made you will need to reference them.

If you are using the variable in the same file that it is created in, you just need to add the name of the variable.

Example: Using "***Int VotePoints = 0;***" in **Player.java,** the same file you made the variable in, you would just type: "**Votepoints**;"

However using "***Int VotePoints = 0;***" in Commands.java, you will need to reference which file that variable is created in, by adding the name in front of it, ***example***: Player.VotePoints.

Since you are not using the integer **VotePoints** in **Player.java** which is where it is set you need to add "**Player.**" to the start of it so it knows to look in the file **Player.java** for the variable "***VotePoints***".

A Real Example:

An example of a variable is "votepoints"

```
public int VotePoints = 0;
```

It's an "int" meaning it's a number and is set by default to 0.

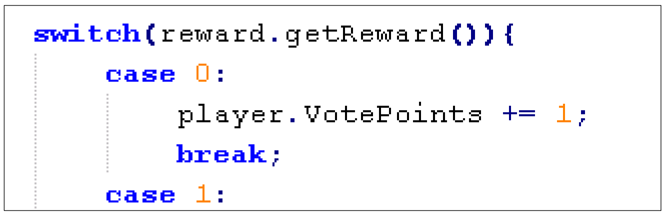
It's created in ***Player.java***.

How you would reference it:

To use ***votepoints*** you need to add the name of the file it is created in, which is “Player”, along with a full stop, then the variables name.

It is used currently in the Poanizer Project Source in ***Commands.java*** in the command “claim”.

“Player.VotePoints”



```
switch(reward.getReward()){
    case 0:
        player.VotePoints += 1;
        break;
    case 1:
```

The votepoints variable is being used in this situation as a reward for voting. If you remember back from the math signs section, you will know that “**+**” means “**add**” or “**plus**”.

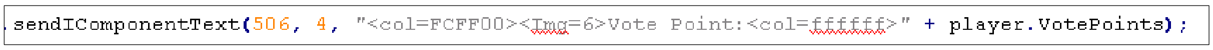
The equal sign is just used as a way of saying “**make the value go up by**”, since if you done just the one “**+**” it means more like “***combine***” rather than “***addition***” if that makes sense.

So when someone does the command “***claim***” and receives the first reward, their “***votepoint***” variable with go up by 1.

<u>How it looks ingame:</u>

An example of the variable being used in game is on the Quest Tab “***Vote Point***” feature.

This feature references the variable but only displays it value and doesn’t change it at all.



```
sendIComponentText(506, 4, "<col=FCFF00><Img=6>Vote Point:<col=ffffff>" + player.VotePoints);
```

Above is the piece of code that displays the variable ingame. As you can see this one only uses one “**+**” and as stated before this generally means to “***combine***” rather then add. Since your combining the number on the end of this text rather than “***adding***” the variable to the total text, which wouldn’t work anyway since text or “***Strings***” aren’t numbers. Hopefully that makes sense.

To make the difference between “***Combine***” and “***Adding***” more clear think of it like this:

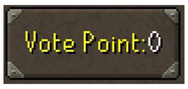
You have 2 integers, a 1 and a 1.

If you go: 1 + 1 **You would get:** 11 since your combining the two 1’s together.

Whereas if you went: 1 += 1 **You would get:** 2 since your adding them together.

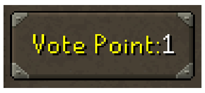
Note: 1 + 1 doesn't equal 11 in coding it was just used to try clarify.

So when they open the quest tab it should show the below image:


Vote Point:0

It shows the text and then the value of the "***VotePoint***" variable after it.

When the user uses the command "***claim***" and gains 1 vote point the value will change by 1 which will change the button on the quest tab to show the new value of that variable:


Vote Point:1

Functions and Arguments

In this part of the guide things will move more towards just how I understand things rather than the proper technical explanation. If you wish to learn more about this section then please refer to the disclaimer at the top and use your own means of research.

Functions

There a different types of functions, but like I said above, I currently don't know much about this part. They are normally defined with statements such as "Void" or any other variable type names.

A function is basically code that starts when the name is initiated. You could think of a function as a machine. When the machine is started it does something. This is the same for Functions. When you start a function it will do the code inside it.

Almost all functions will have parentheses after them, **()**. This is a good way to know if something is a variable or a function.

The basic layout for a function is:

Name(Arguments) {

Code

};

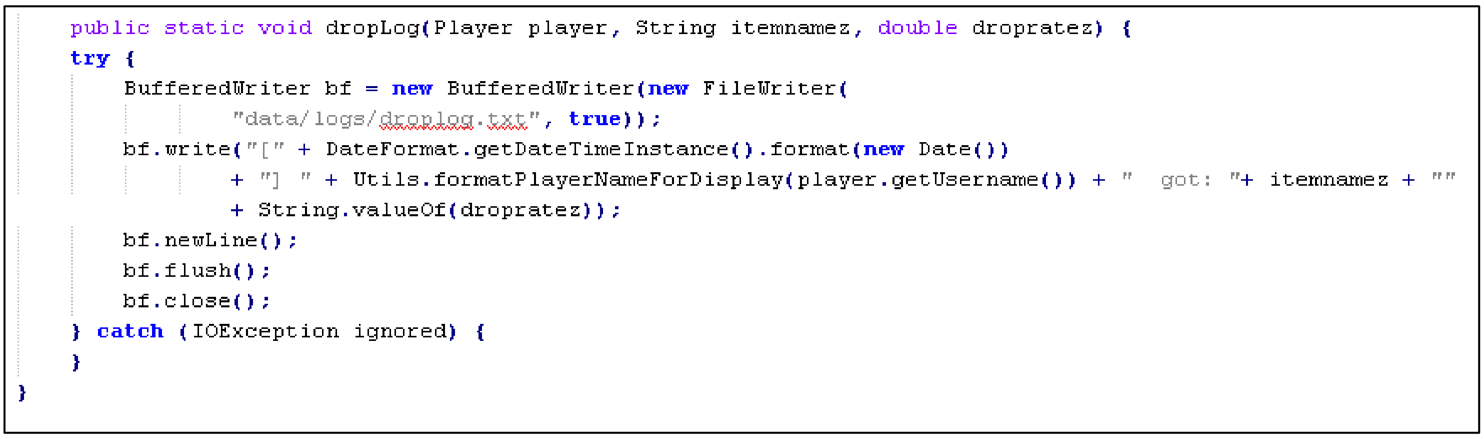
The main thing to know about functions and it will really be the only thing I will try explain in this point is how the **Arguments** work.

Arguments

Arguments are the variables that the function requires inside its parentheses. Refering back to the machine metaphor, Arguments are like the fuel. Some machines require fuel to make them run, if you put the wrong fuel into the machine it breaks.

These arguments will normally be variables so they will be look like "***Int Number***" or **"String Text".** What this means, if we refer back to the variables section, is that the function requires an integer, which it will set to the variable "***Number***" and a String, which it will set to the variable "***Text***".

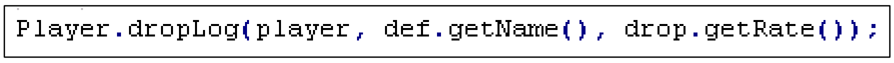
For this example, we will use the **dropLog** function in **Player.Java.**



```
    public static void dropLog(Player player, String itemnamez, double dropratez) {
    try {
        BufferedWriter bf = new BufferedWriter(new FileWriter(
                "data/logs/droplog.txt", true));
        bf.write("[" + DateFormat.getDateTimeInstance().format(new Date())
                + "]  " + Utils.formatPlayerNameForDisplay(player.getUsername()) + "  got: "+ itemnamez + ""
                + String.valueOf(dropratez));
        bf.newLine();
        bf.flush();
        bf.close();
    } catch (IOException ignored) {
    }
}
```

As you can see in the screenshot above, the function is called "***dropLog***", and it requires a Player (Entity) which it sets to the variable "***player***", a String which it will set to "***itemnamez***" and a Double which it sets to "***dropratez***".

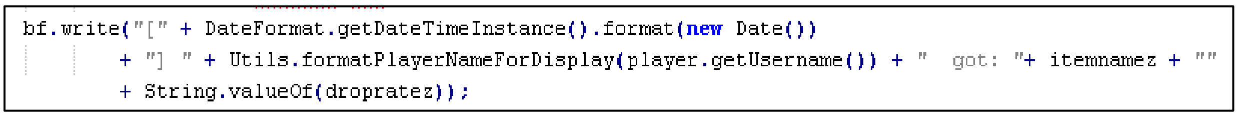
To initiate (Start) the Function "***dropLog***" you must start it by using the below piece of code.



```
Player.dropLog(player, def.getName(), drop.getRate());
```

Initiating functions uses the same layout as referencing Variables. You start by typing the file name that the function is created in, for this case you use "***Player***" since it was created in "***Player.java***" Then you add the name of the function "***dropLogs"*** along with its arguments inside the parentheses.

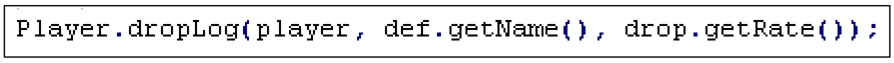
After the closed parentheses, **)** , of the function itself, not the code that starts it**,** it has curly brackets **{ }** , there is code in between them. This code will run when the function is initiated. In this case it will write to the file, which is also the location: “***data/logs/droplog.txt***”**.**



```
bf.write("[" + DateFormat.getDateTimeInstance().format(new Date())
        + "] " + Utils.formatPlayerNameForDisplay(player.getUsername()) + "  got: "+ itemnamez + ""
        + String.valueOf(dropratez));
```

If you examine the variables inside the code that the function runs, you will notice they match up with variables in the arguments that initiated the function

Example: On the line that starts with “***bf.write***” (image above), you will see that there are variables and functions used in the code: “***Player.getUsername()***”, “***itemnamez”***”, “***dropratez***”. These are the same variables that were in the code that initiated the function (Image below).



```
Player.dropLog(player, def.getName(), drop.getRate());
```

Another Explanation:

So calling a function is the same as calling a variable.

To call a function you use the same method as calling a variable. If you using it in a different than the one it is created in, add the file name, if not just use the functions name on its own, no “FILE.” required.

Then add the arguments in the parentheses **( )** .

If it has arguments then you will need to send them, in matching type and order

Example: FunctionName**(** Arguments**)** ;

If the function itself doesn’t take any arguments you leave the parentheses blank .

Example: FunctionName**()** ;

Another Explanation Example:

Files used: Trade.java

Partner and ***Trader*** are just different ways renaming the file "***player***"

If you notice you won't find "**getTrade()**" function in **Trade.java**

Anything that starts with "**player**." or any other changed entity of that will be referencing something in the file... **player.java**

Example: partner.getTrade()

There is no "***getTrade()***" in Trade.java but if you look in ***Player.java*** you will find....

```
public Trade getTrade() {
  return trade;
 }
```

What im trying to say is if the function is created in trade.java or the file you editing, you won't need to add ***player.*** to the start of it since your referencing it from the same file it is created in.

If you type the name and the arguments in the brackets () then it will do everything inside the brackets of the main function { } instead of you having to type it all out again

<u>What does this mean?</u>

Basically in summary this means that all the variables required for the function to work properly are found inside the parentheses. When you want to start a function you need to add variables inside the parentheses of the starting code that are the same variable type and in the same order as the ones in the parentheses of the function

<u>Example</u>:

Each variable has been colour coded to match its partner. The things left in black are the type of variable the function requires. The variables in the Start Code are those types.

Referencing the Function: Player.**dropLog**(**player**, **def.getName()**, **drop.getRate()**);

Function's Code: **dropLog(Player player , String itemnamez, double dropratez)**

"***player***" is an entity, which is what the function requests with "**Player**".

"***def.getName()***" is a function that prints the Name in a String. "**String**"

"**drop.getRate()**" is a function that prints the drop rate in a decimal. "**Double**"

That's all the ways I can think of to try explain how functions work. If you still do not understand try looking for answers online. Use searches such as "***Java Functions***" or "***Java Arguments***".

Imports

Yet again this section I do not know much about them so this is my interpretation of them.

What are Imports?

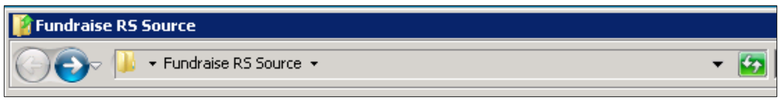
```
import com.rs.game.player.Kilner;
import com.rs.game.player.Player;
import com.rs.game.player.MusicsManager;
import com.rs.game.player.Skills;
import com.rs.game.player.actions.Summoning;
import com.rs.game.player.content.Notes.Note;
import com.rs.game.player.content.slayer.SlayerTasks;
```

Imports are the ways you link files together. They allow you to have access to any "***Variables***" or "***Functions***" that were created in the file you're importing.

To import a file you need to type the word "***import***" first to tell the server that you're creating an import. After you have typed that you just add the path to the file from the folder called "***src***" which is location in your main Source Folder.

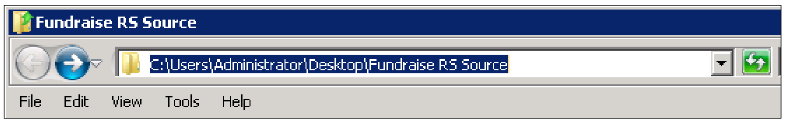
What is a Path?

The path is just the navigation required to get to a certain folder or file. If you open up your servers "***Source***" folder, you will see its current path up the top of your window.



The above picture just has the Source Folders name because it is located on the desktop. If you compare this path to that of an import, the downward arrow in the above picture would be replaced by a dot "**.**".

If you click in the blank space to the right of the current path, it will show the path in "**Windows"** format.



Now instead of just saying "***Fundraise RS Source***" It says "***C:\Users\Adminstrator\Desktop\Fundraise RS Source***"

This is the actual destination of the Source folder located on your computer. You may notice this style of path is what your "***Compiler***" uses to find your "***JDK***".

```
@echo off
"C:\Program Files (x86)\Java\jdk1.7.0_25\bin\javac.exe"
echo compiled Player folder
```

Back to Imports

So taking from what was discussed above. The imports have the same order of folders that lead to the location of the file as it would if you opened each one in Windows Explorer.

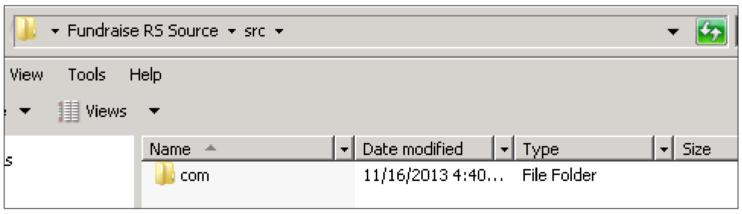
Each name is a name of a folder separated by a dot instead of a slash "\".

Example:

For this example we will be using the import for "***Player.java***"

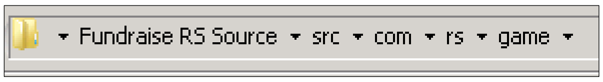
```
import com.rs.game.player.Player;
```

If you go into our Source folder and open "***src***"we will see a folder named "***com***".



Notice above the folder, the path shows: “***Fundraise RS Source*** · ***src***”

If you navigate through the folders “***com***” then ”***rs***“ then “***game***” and finally “***player***” you will notice your path has changed at the top.



This path matches the folders you just went through and the names at the start of the import for “***Player.java”:*** *com · rs · game · player = com.rs.game.player*

```
import com.rs.game.player.Player;
```

The last word on the end of the import is the name of the file. Generally it will have a capital.

If you look in the “***Player***” folder that you just opened you will see a file called “**Player**”. The name of this file and its location matches the path in the import, so this is the file that the import will be connecting to. Therefore any “***Variable***” or “***Function***” created in the file that’s being imported can be referenced in the file that the **Import** was created in.

Refer back to “***Calling/Referencing Variables”*** to know the structure of referencing variables from different file than the one it was created in.

Java Imports

Java imports always start with “***java***”. They are features built into the java code that can be used in your Runescape Private server. Normally they are for the technical side of things or for doing Non Runescape related coding.

```
import java.io.File;
import java.io.FileWriter;
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.FileWriter;
```

These imports and features can be handy to use when making technical code. However as far as I know you will either need to find them on the internet or know a bit about java to know which features are available to import.

Examples of Java Features:

System.currentTimeMillis(): Gets your computer's current time in milliseconds.

bf.write(): Writes something into pre-created document

Calendar.getInstance(): Gets the current Calendar Details.

Common RSPS Errors.

Errors Suck. But it's alright, everyone makes them and I will discuss some of the common errors that I see people asking about.

Do remember if your using the default **667** compiler and you get an error it will repeat the error a few times, since you are compiling more than once in the one .Bat

How to Read Errors

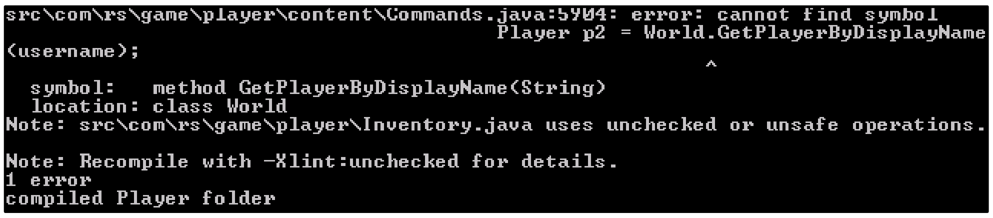
The first line of the error will show you the path to location of the file that the error occurred in and what line. It will also say what the error was.

It will then have an up facing arrow on the point where the code broke at. This point isn't always accurate since if the error is due to you not closing a bracket the compiler will keep reading through the file assuming everything after the broken bracket are "***arguments***" or code in general.

Below this it will then show information about the error. Normally the next line will depend on what type of error you will have but then the line after, will be the Location of the error.

This information will be repeated for each error that is found. After each error has been stated it will display the total amount of errors.

Example:



```
src\com\rs\game\player\content\Commands.java:5904: error: cannot find symbol
                                        Player p2 = World.GetPlayerByDisplayName
(username);
                                                          ^
  symbol:   method GetPlayerByDisplayName(String)
  location: class World
Note: src\com\rs\game\player\Inventory.java uses unchecked or unsafe operations.

Note: Recompile with -Xlint:unchecked for details.
1 error
compiled Player folder
```

Location: *“src\com\rs\game\player\content\Commands.java”*

Line*: 5904*

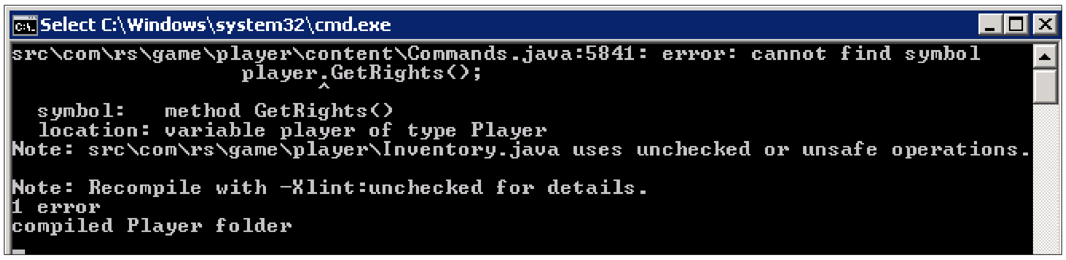
Error: *Cannot find the symbol.*

Depending on the type: *Symbol: method GetPlayerByDisplayName(String)* – **That is the symbol it cant find.**

Location: The Java Class*: “World”*

How many errors Total: *1 error*

Cannot find the symbol



Select C:\Windows\system32\cmd.exe

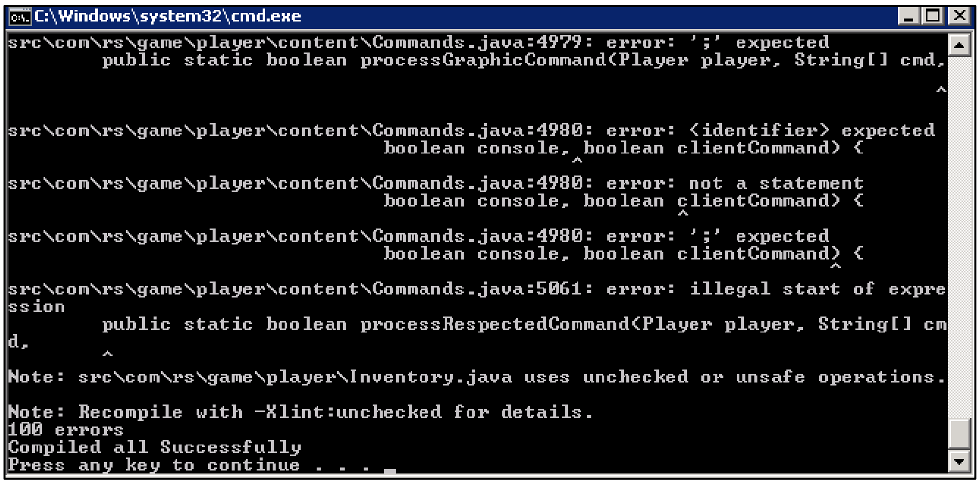
```
src\com\rs\game\player\content\Commands.java:5841: error: cannot find symbol
                  player.GetRights();
                        ^
  symbol:   method GetRights()
  location: variable player of type Player
Note: src\com\rs\game\player\Inventory.java uses unchecked or unsafe operations.

Note: Recompile with -Xlint:unchecked for details.
1 error
compiled Player folder
```

This error is one of the most common ones for new coders and can have multiple reasons why it is occurring.

Either:

- You have spelt a function/Variable incorrectly. Spelling or Case Sensitive.
- You have not added an “***import***” for the file the Variable/Function is located in.

100 Errors



```
src\com\rs\game\player\content\Commands.java:4979: error: ';' expected
        public static boolean processGraphicCommand(Player player, String[] cmd,

                                                                                ^
src\com\rs\game\player\content\Commands.java:4980: error: <identifier> expected
                                boolean console, boolean clientCommand) {
                                                ^
src\com\rs\game\player\content\Commands.java:4980: error: not a statement
                                boolean console, boolean clientCommand) {
                                                         ^
src\com\rs\game\player\content\Commands.java:4980: error: ';' expected
                                boolean console, boolean clientCommand) {
                                                                      ^
src\com\rs\game\player\content\Commands.java:5061: error: illegal start of expre
ssion
        public static boolean processRespectedCommand(Player player, String[] cm
d,
        ^
Note: src\com\rs\game\player\Inventory.java uses unchecked or unsafe operations.

Note: Recompile with -Xlint:unchecked for details.
100 errors
Compiled all Successfully
Press any key to continue . . . _
```

This one is the most annoying Error in my opinion. It can mean ages of slow hunting through code.

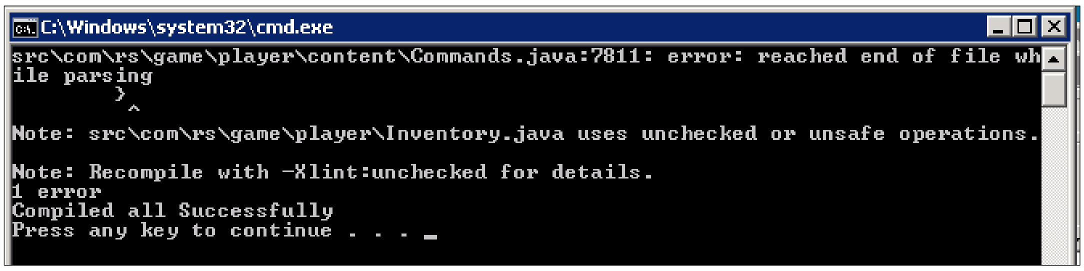
Reason it happens:

Generally you will get this error when you have not closed a bracket: **)** or **}**

Referring back the syntax side of this guide, each open bracket **(** or **{** needs a closing bracket **)** or **}** .

If each open one does not have a closing one then the compiler will keep reading everything after it thinking that it's all part of what was meant to be inside the brackets and it ends up breaking everything

Reached end of file while parsing

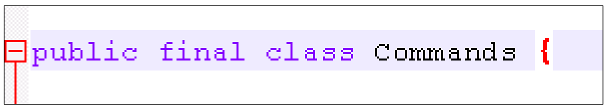


```
src\com\rs\game\player\content\Commands.java:7811: error: reached end of file wh
ile parsing
        }
         ^
Note: src\com\rs\game\player\Inventory.java uses unchecked or unsafe operations.

Note: Recompile with -Xlint:unchecked for details.
1 error
Compiled all Successfully
Press any key to continue . . . _
```

This error I'm not 100% sure on, but it's basically the same as the "***100 errors***" Error.

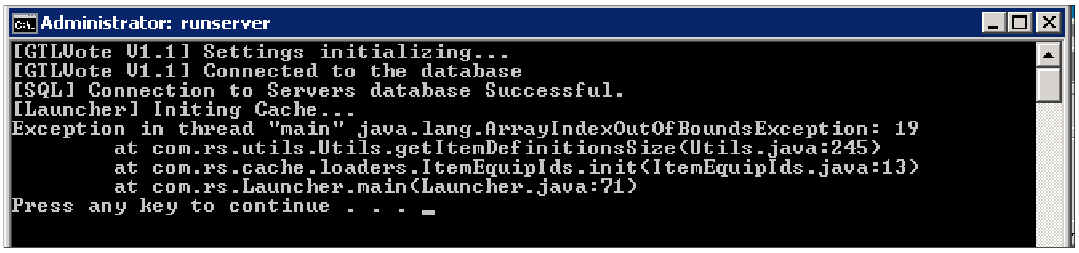
What I understand about this error is that it has reached the end of the file without finding the closing bracket for the main class.

Each .Java file will have a main class name that is normally found under the imports. The closing bracket **}** for this is generally found at the very bottom of the file. Meaning it is the very last bracket. So if it can't find the end bracket, then it "***Reached the end***" of the file without finding it.



```
public final class Commands {
```

Exception in thread main

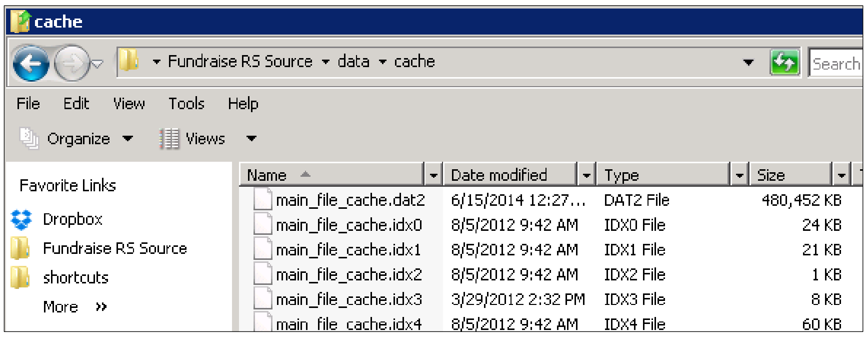


```
[GTLVote V1.1] Settings initializing...
[GTLVote V1.1] Connected to the database
[SQL] Connection to Servers database Successful.
[Launcher] Initing Cache...
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: 19
        at com.rs.utils.Utils.getItemDefinitionsSize(Utils.java:245)
        at com.rs.cache.loaders.ItemEquipIds.init(ItemEquipIds.java:13)
        at com.rs.Launcher.main(Launcher.java:71)
Press any key to continue . . .
```

This error can mean multiple things I think. However the reason most people ask about is because the server cannot find the cache.

To fix it:

Add a full compatible cache into the “***data/cache***” folder.



Name	Date modified	Type	Size
main_file_cache.dat2	6/15/2014 12:27…	DAT2 File	480,452 KB
main_file_cache.idx0	8/5/2012 9:42 AM	IDX0 File	24 KB
main_file_cache.idx1	8/5/2012 9:42 AM	IDX1 File	21 KB
main_file_cache.idx2	8/5/2012 9:42 AM	IDX2 File	1 KB
main_file_cache.idx3	3/29/2012 2:32 PM	IDX3 File	8 KB
main_file_cache.idx4	8/5/2012 9:42 AM	IDX4 File	60 KB

Null Pointer Exception

This error is quite technical for my knowledge and I did search the internet to know what it means. I have a rough idea but you will find it probably more helpful to find a better explanation.

But what I got from it is a Null Pointer Exception means that something is pointing to null, which gives you and error. Null is another word for nothing.

So what I understand is that when you reference something, you’re pointing to that certain Variable/Function to get its value. If the value doesn’t exist or something is wrong and it returns “Null” when it is expecting something else, then it will break.

Simple Explanation: you're missing something, or a value is equal to nothing when it should have a value.

How to Port Forward

- Site with all routers
- Explanation of 10.1.1.3 or Ipv4
- How to check if open

When you open a RSPS you need to open a port on your computer that the information can be sent and received through. Normally your Router/Modem will block most ports not allowing other people to connect to your RSPS. The general port used in RSPS is ***43594.*** Port forwarding will also be different for each brand and possibly model of Modem/Router you own.

A good site to find out how to port forward your router is: **http://portforward.com/**

They have a large variety of different Brands and Models with tutorials showing you how to do it.

I only just figured out how to port forward recently so I will discuss some things you may need to understand first briefly below. Remember again this will just be my knowledge of it so it may not be 100% accurate. If something isn't fully explained in here it is best to use the internet to find more information.

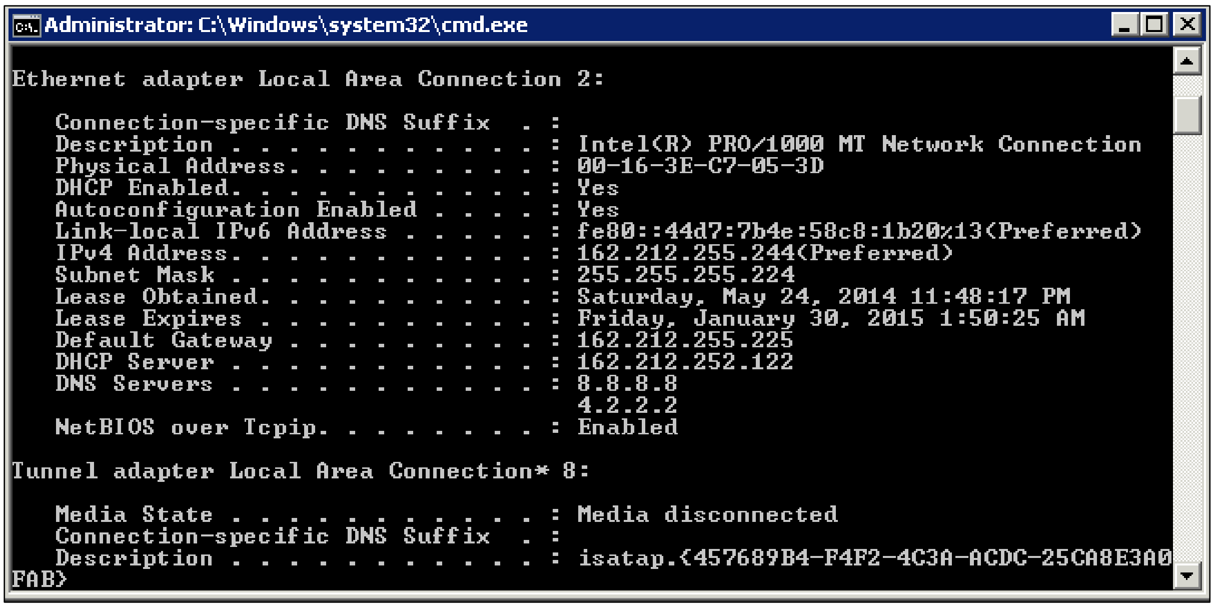
Ipconfig

Ipconfig is a command prompt command used to display all your network information. It will show all the different IP's used. To display this information, open a command prompt window and type: "***Ipconfig /all*** "

Physical Address: This is also known as your MAC address, this is how your device is identified, as this address generally won't be on any other device.

Ipv4: this is the address that your Modem will identify you as.

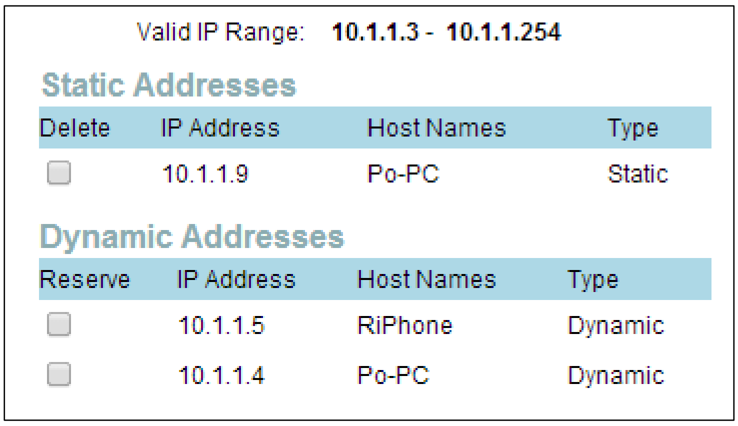
Default Gateway: This is the IP address you generally use to connect to the Modem from your computer this may change depending on your brand/model. ***Example:*** 10.1.1.1



```
Ethernet adapter Local Area Connection 2:

   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : Intel(R) PRO/1000 MT Network Connection
   Physical Address. . . . . . . . . : 00-16-3E-C7-05-3D
   DHCP Enabled. . . . . . . . . . . : Yes
   Autoconfiguration Enabled . . . . : Yes
   Link-local IPv6 Address . . . . . : fe80::44d7:7b4e:58c8:1b20%13(Preferred)
   IPv4 Address. . . . . . . . . . . : 162.212.255.244(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.255.224
   Lease Obtained. . . . . . . . . . : Saturday, May 24, 2014 11:48:17 PM
   Lease Expires . . . . . . . . . . : Friday, January 30, 2015 1:50:25 AM
   Default Gateway . . . . . . . . . : 162.212.255.225
   DHCP Server . . . . . . . . . . . : 162.212.252.122
   DNS Servers . . . . . . . . . . . : 8.8.8.8
                                       4.2.2.2
   NetBIOS over Tcpip. . . . . . . . : Enabled

Tunnel adapter Local Area Connection* 8:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : isatap.{457689B4-F4F2-4C3A-ACDC-25CA8E3A0
FAB}
```

One thing to remember is your Ipv4 address can change if it is dynamic. If it is set to static then it will be fine.

If I check my router it will show all the available Ipv4 addresses for each device connected to it along with the hostname and whether it is static or dynamic. The below image could be kind of confusing since I have 2 computers named "***PO-PC***". But you will see their addresses are different.



Valid IP Range: **10.1.1.3 - 10.1.1.254**

Static Addresses

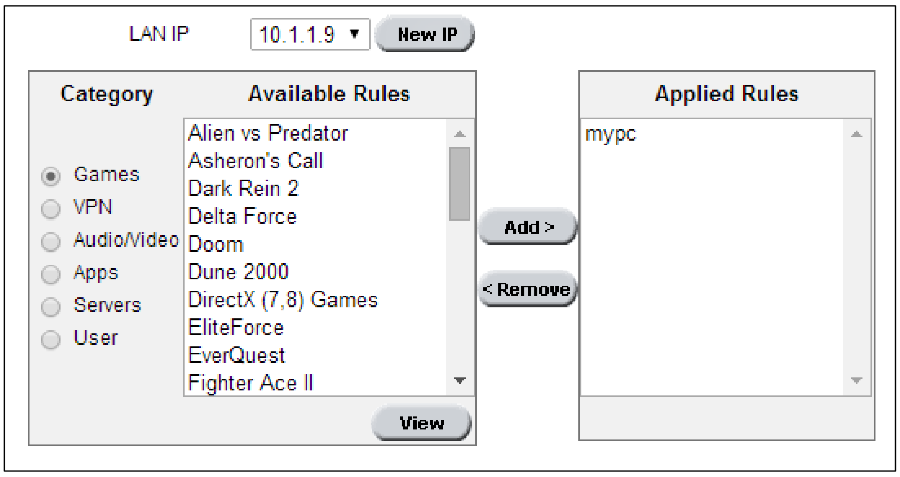
Delete	IP Address	Host Names	Type
☐	10.1.1.9	Po-PC	Static

Dynamic Addresses

Reserve	IP Address	Host Names	Type
☐	10.1.1.5	RiPhone	Dynamic
☐	10.1.1.4	Po-PC	Dynamic

Some routers will only allow the port you open to be accessed by certain devices. It will use your devices Ipv4 for this.

The below example is for my D-Link Modem. The box labelled "***LAN IP***" is which Ipv4 address the port or "Rule" will be open for. The box on the right "***Applied Rules***" are the current Rules/Ports enabled for the IPv4 Address "10.1.1.9".


LAN IP 10.1.1.9 New IP

Category: [✓ Games] VPN Audio/Video Apps Servers User

Available Rules: Alien vs Predator, Asheron's Call, Dark Rein 2, Delta Force, Doom, Dune 2000, DirectX (7,8) Games, EliteForce, EverQuest, Fighter Ace II

Applied Rules: mypc

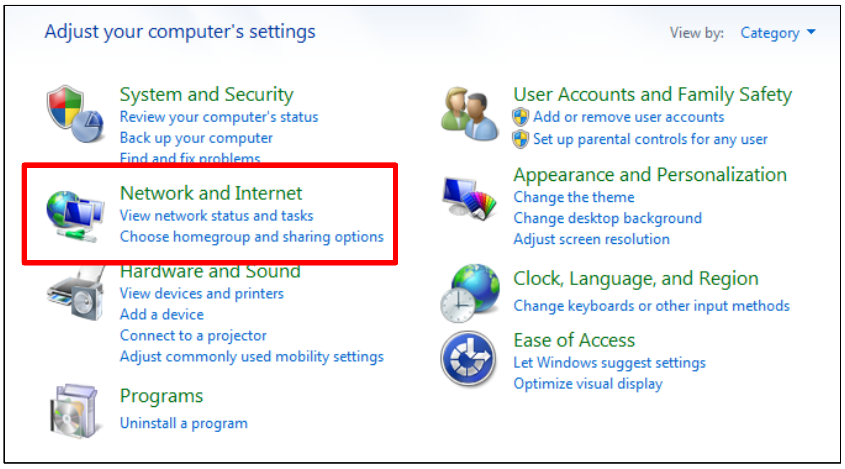
Making your IPv4 Static

To make your Ipv4 static you will need to edit some of your settings in control panel.

You will need Administrative control and I recommend only doing this if you really need to. Sometimes your Ipv4 will change if you swap Networks (Go to a friend's house, etc). When this happens and you have set your Ip to be static on a certain IP, the new Modem may reject your computer because it formats the Ip addresses differently.

This is for Windows 7 and may vary depending on the computer.

Go to Control Panel and click on "Network and Internet".


Adjust your computer's settings

View by: Category

System and Security
Review your computer's status
Back up your computer
Find and fix problems

Network and Internet
View network status and tasks
Choose homegroup and sharing options

Hardware and Sound
View devices and printers
Add a device
Connect to a projector
Adjust commonly used mobility settings

Programs
Uninstall a program

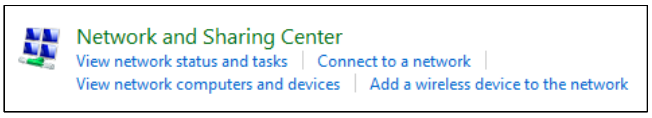
User Accounts and Family Safety
Add or remove user accounts
Set up parental controls for any user

Appearance and Personalization
Change the theme
Change desktop background
Adjust screen resolution

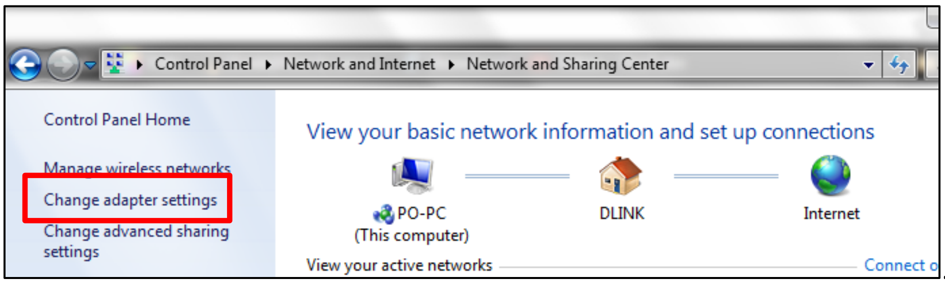
Clock, Language, and Region
Change keyboards or other input methods

Ease of Access
Let Windows suggest settings
Optimize visual display

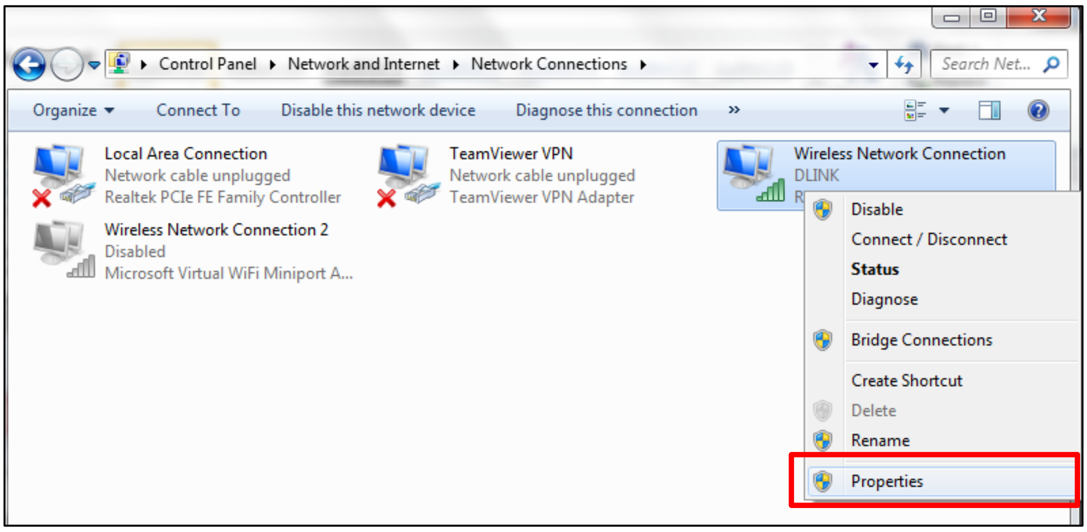
Open the "Network and Sharing Center".


Network and Sharing Center
View network status and tasks
Connect to a network
View network computers and devices
Add a wireless device to the network

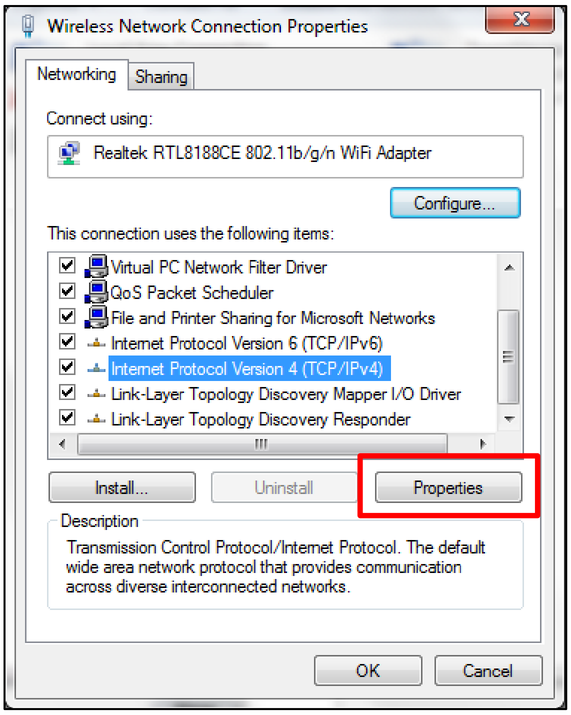
Then Click “Change adapter Settings”


Control Panel
Network and Internet
Network and Sharing Center
Control Panel Home
Manage wireless networks
Change adapter settings
Change advanced sharing settings
View your basic network information and set up connections
PO-PC
(This computer)
DLINK
Internet
View your active networks
Connect o

Right Click your Network Card or Ethernet Port you are using to connect to the internet. Then click on “***Properties***”



Scroll down and click on “Internet Protocol Version 4”, then select properties.



Wireless Network Connection Properties

Networking | Sharing

Connect using:

Realtek RTL8188CE 802.11b/g/n WiFi Adapter

Configure...

This connection uses the following items:

- [✓] Virtual PC Network Filter Driver
- [✓] QoS Packet Scheduler
- [✓] File and Printer Sharing for Microsoft Networks
- [✓] Internet Protocol Version 6 (TCP/IPv6)
- [✓] Internet Protocol Version 4 (TCP/IPv4)
- [✓] Link-Layer Topology Discovery Mapper I/O Driver
- [✓] Link-Layer Topology Discovery Responder

Install... | Uninstall | Properties

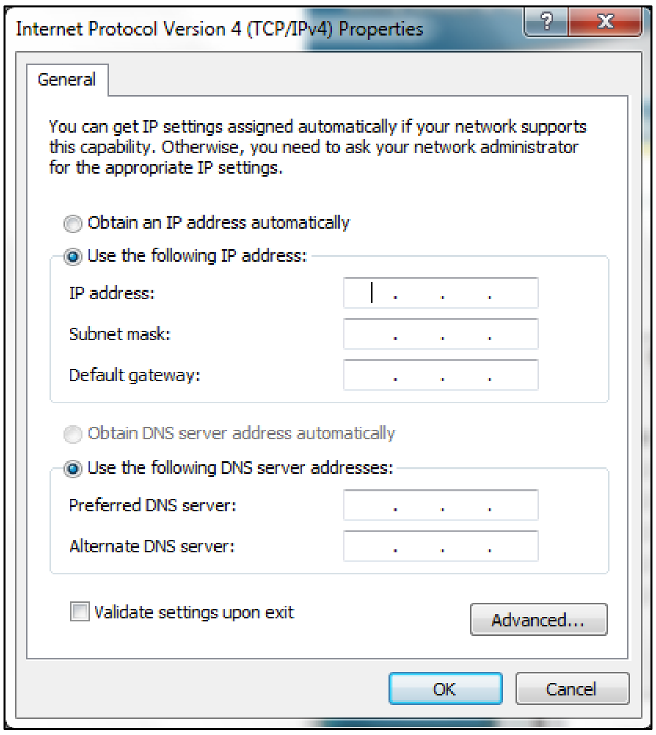
Description

Transmission Control Protocol/Internet Protocol. The default wide area network protocol that provides communication across diverse interconnected networks.

OK | Cancel

In this box you will have 2 options. "***Obtain X Automatically***" or "***use the following X***".

It should be set to obtain it automatically on both. However if you want to set a static Ip then you must enter in the Information using the option "**Use the following**".



Internet Protocol Version 4 (TCP/IPv4) Properties

General

You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

- ○ Obtain an IP address automatically
- [✓ ●] Use the following IP address:

IP address: . . .

Subnet mask: . . .

Default gateway: . . .

- ○ Obtain DNS server address automatically
- [✓ ●] Use the following DNS server addresses:

Preferred DNS server: . . .

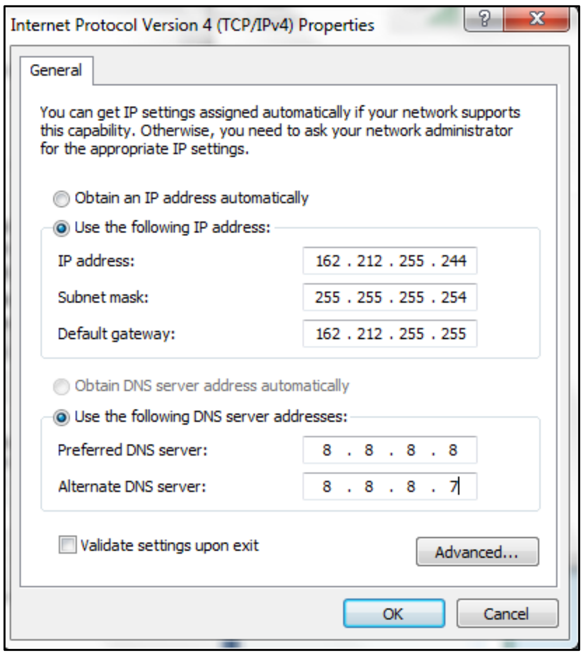
Alternate DNS server: . . .

[ ] Validate settings upon exit

Advanced...

OK | Cancel

Match all the information to the information displayed when using the “Ipconfig” Command. Honestly I’m not 100% sure on the alternative DNS server, so normally I just do either 1 more or 1 less. If we are using the same Ipconfig that was shown above our inputs should look like this:


Internet Protocol Version 4 (TCP/IPv4) Properties

General

You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

○ Obtain an IP address automatically
● Use the following IP address:

IP address:	162 . 212 . 255 . 244
Subnet mask:	255 . 255 . 255 . 254
Default gateway:	162 . 212 . 255 . 255

○ Obtain DNS server address automatically
● Use the following DNS server addresses:

Preferred DNS server:	8 . 8 . 8 . 8
Alternate DNS server:	8 . 8 . 8 . 7

☐ Validate settings upon exit

How to setup a Hamachi Server

Why choose a Hamachi Network?

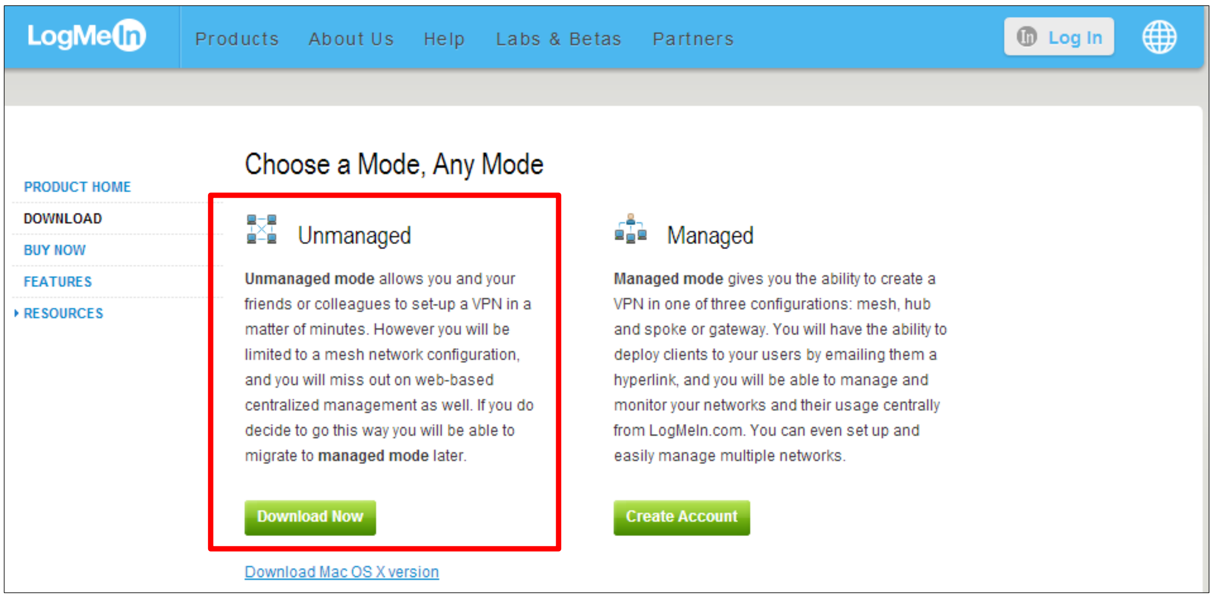
Hamachi also known as “***LogMeIn***” is a 3rd party programme that allows connections between “***Clients***” in a virtual network that the software creates. Hamachi is good to use if you cannot afford or do not wish to use a VPS but can’t get port forwarding working.

How to set up a Hamachi Network

Step One:

Head to: https://secure.logmein.com/products/hamachi/download.aspx

And download the “***UnManaged***” Version. I’m not sure on what the difference is, but this one works.


Choose a Mode, Any Mode

Unmanaged

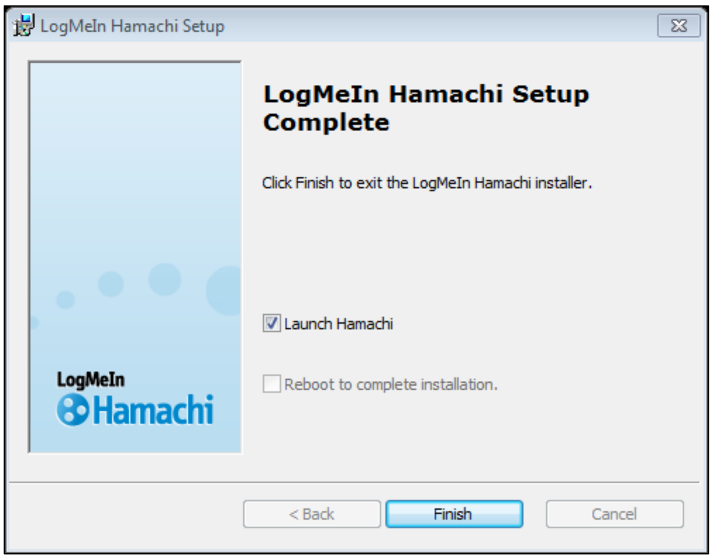
Unmanaged mode allows you and your friends or colleagues to set-up a VPN in a matter of minutes. However you will be limited to a mesh network configuration, and you will miss out on web-based centralized management as well. If you do decide to go this way you will be able to migrate to **managed mode** later.

Managed

Managed mode gives you the ability to create a VPN in one of three configurations: mesh, hub and spoke or gateway. You will have the ability to deploy clients to your users by emailing them a hyperlink, and you will be able to manage and monitor your networks and their usage centrally from LogMeIn.com. You can even set up and easily manage multiple networks.

Step Two:

When that is finished downloading, Install it.


LogMeIn Hamachi Setup

LogMeIn Hamachi Setup Complete

Click Finish to exit the LogMeIn Hamachi installer.

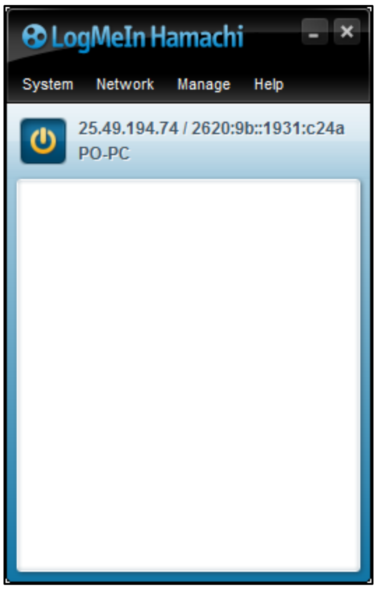
[✓] Launch Hamachi

[ ] Reboot to complete installation.

Step Three:

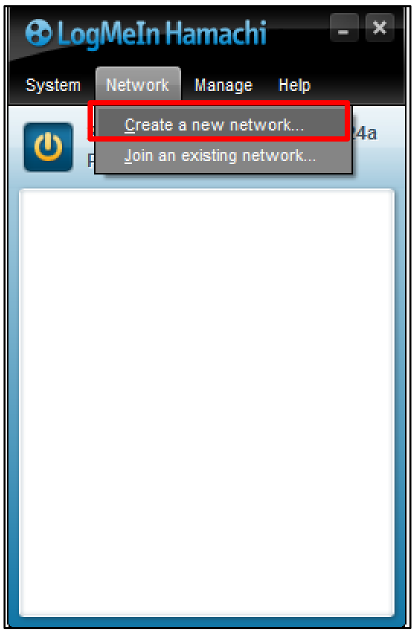
Once it opens you should see the below interface.

The numbers next to the power button are your IPv4 and IPv6 Hamachi Addresses, along with your computer's name below. Mine is "***PO-PC***" .



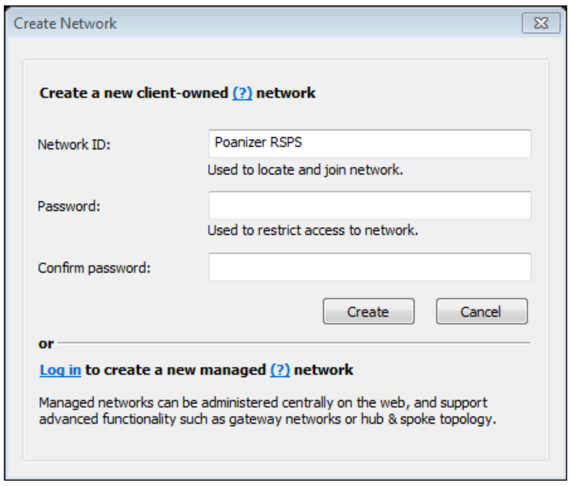
Step Four:

Click the network tab and click "***Create a new Network***"



Step Five:

Fill in your server details. The name will be the name people enter to connect to your network. It will be Case Sensitive. The password is the password required to join, this is also Case Sensitive.


Create Network

Create a new client-owned (?) network

Network ID: Poanizer RSPS
Used to locate and join network.

Password:
Used to restrict access to network.

Confirm password:

Create Cancel

or

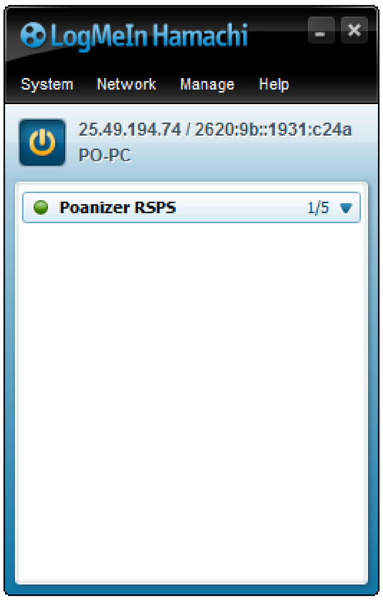
Log in to create a new managed (?) network

Managed networks can be administered centrally on the web, and support advanced functionality such as gateway networks or hub & spoke topology.

Step Six:

You will now see your Network appear below your IP's. You will see on the ***Free*** version of Hamachi you can only hold 5 people in 1 network. If you are opening a Public RSPS I would suggest you make 2-3 Networks with similar names.

Example: "Poanizer RSPS", "Poanizer RSPS2", "Poanizer RSPS3"



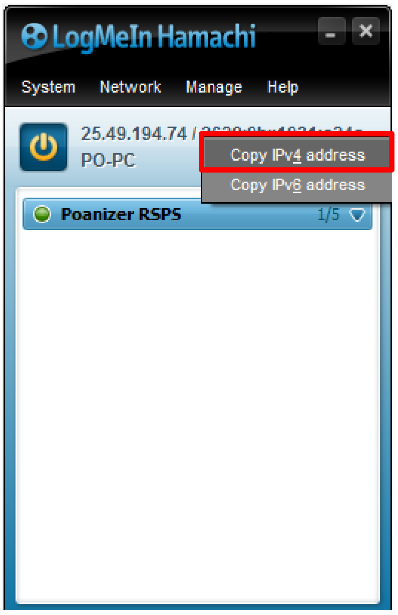
25.49.194.74 / 2620:9b::1931:c24a
PO-PC

Poanizer RSPS 1/5

Step Seven:

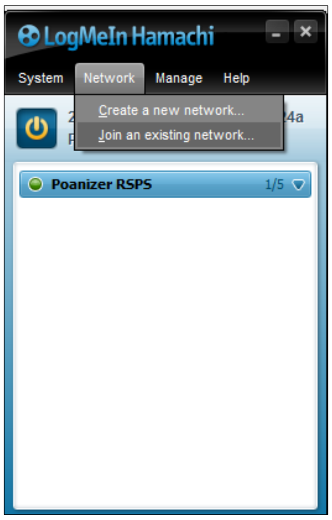
Now you will need to set your Hamachi IPv4 address as the IP in the client (Refer to Getting Started Section). To get your IPv4 Address, right click the numbers and click "***Copy IPv4 Address***" Then when you're in the client's ***Loader.java*** push ***Ctrl + V*** or use ***Paste*** To add your IPv4.

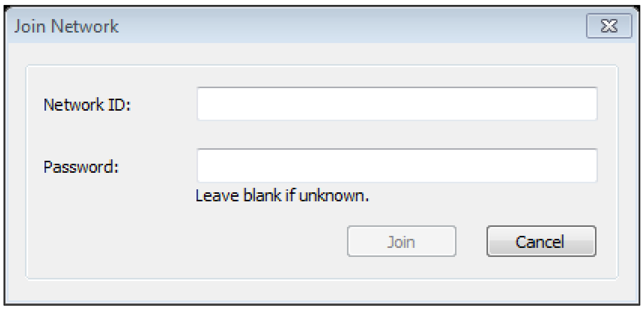
My IPv4: ***25.49.194.74***



How to Join a Hamachi Network

This is very easy. All you need to do is click "***Network***" and click ***Join an Existing Network*** then enter the networks details.




Join Network

Network ID:

Password:

Leave blank if unknown.

Virtual Private Servers - VPS

A VPS is a Virtual Private Server.

VPS’s are practically a 2nd computer that you connect to remotely and are able to control.

If you have ever used the programme TeamViewer before, then the VPS experience is pretty much the same, except nobody is on the other computer.

Most VPS's will stay online 24/7 regardless whether you close the connection between your computer and the VPS. Sometimes you may find your VPS has randomly restarted though.

All VPS's are port forwarded. You just need to allow the port through the firewall.

What you need to know about VPS's

First off, you do **NOT** need a VPS.

I do encourage people to use a VPS when creating a server however you do not need a VPS to run your own Server.

The alternative to using a VPS is to Port Forward your own computer or use Hamachi.

However a VPS is a lot more secure in my opinion, except for the fact that you can lose ALL your files, if not backed up when the VPS runs out.

If you are looking to purchase a VPS I also recommend you look for a “***DDos Protected***” one. If you run your server locally people can decompile your client and find your home ip address.

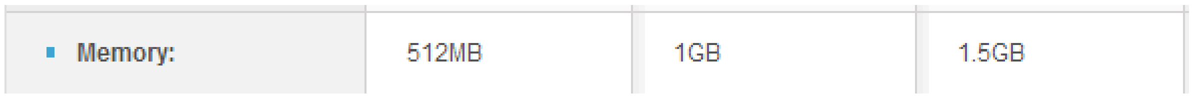
-**Note**-

This next section may vary depending on the Server's Source and the VPS Company.
It is also very opinion based, VPS results may vary. Make your own decision.

Minimum VPS Requirements:

RAM/Memory:

The memory of your VPS is quite important.



▪ Memory:	512MB	1GB	1.5GB

512mb: This will most likely be the bare minimum required.

1gb: Should run a new server fairly well.

2gb: This would be a good amount to run smoothly, or 20+ players.

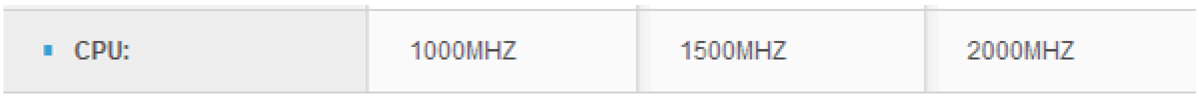
3gb: Anything over 3GB RAM is only needed for big servers.

CPU:

CPU speed is not to important.

Normally CPU is matched with RAM. Example:2GB RAM = 2Ghz CPU

*** 1000Mhz = 1Ghz ***



▪ CPU:	1000MHZ	1500MHZ	2000MHZ

1Ghz: Decent starting out CPU but may lag.

1.5Ghz: May make a small amount of difference.

2Ghz: Should run smoothly.

3Ghz: Helpful to use if coding on VPS.

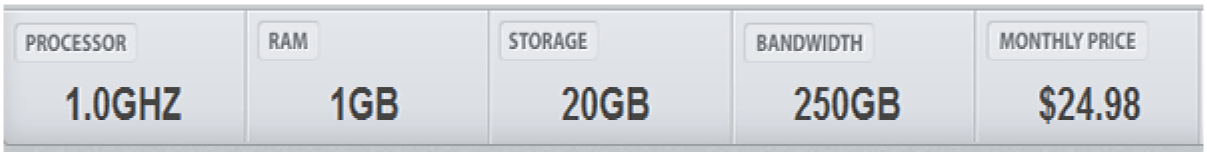
What is the Best VPS Company

There will be a lot of varied opinions on which are the best VPS Companies. But here's my take on the 3 I have used.

VPS Land:

VPS Land was my first VPS Company. I would not recommend them. They give you what you asked for, a VPS but nothing else really.

The customer support was terrible, they took 24+ Hours to reply to any support ticket I submitted. The storage available on their VPS's is also fairly small and their pricing is way too expensive also in my personal opinion.



PROCESSOR	RAM	STORAGE	BANDWIDTH	MONTHLY PRICE
1.0GHZ	1GB	20GB	250GB	$24.98

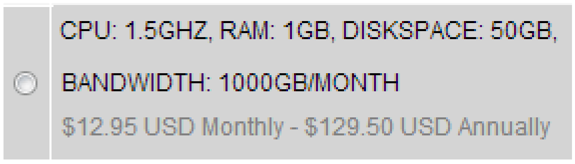
Taken from http://www.vpsland.com/windows_vps.php (27/10/2013)

BurstNet:

Burst Net was my 2nd VPS Company.

I do think they are quite a good VPS company. They had Excellent customer support. Ticket replies were from 5 minutes to 2 hours.

The Specs to Price ratio is very cheap. 10$ cheaper than VPS Land



CPU: 1.5GHZ, RAM: 1GB, DISKSPACE: 50GB, BANDWIDTH: 1000GB/MONTH

$12.95 USD Monthly - $129.50 USD Annually

Taken from: https://service.burst.net/cart.php?gid=5 **(27/10/2013)**

However…

I did disconnect a fair amount when I first signed up. The problem was I needed to "set my ip" which they ended up doing for me after I submitted a support ticket.

I also came across a big problem when they updated their system. It claimed I was "trying to commit fraud" and wouldn't let me access my VPS.

On purchase of a VPS, if you get the mobile number used for confirmation wrong 3 times you get labelled as a Fraud and must fill out a form with all your personal information on it to be able to repurchase a VPS.

This put me away from BurstNet and so my opinion on them now is very low.

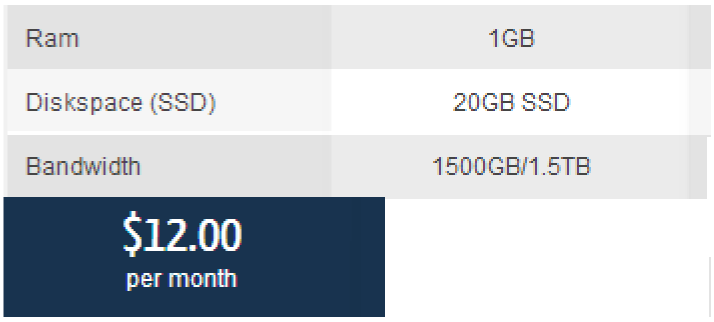
TrentaHost:

TrentaHost

TrentaHost was my 3rd VPS Company and the one I stayed with.

I have not had any problems with them. The Customer support is good. Average reply is 1-4 hours if not peak in time Prices are good with fairly cheap efficient DDOS Protection. You can also find good discount codes on their Facebook page which bring the price down quite a bit.

I have also noticed that every New VPS is setup and ready after around 30 minutes of purchase.



Ram	1GB
Diskspace (SSD)	20GB SSD
Bandwidth	1500GB/1.5TB

$12.00
per month

Taken from https://trentahost.com/budget-windows-vps/ **(27/10/2013)**

Allowing a port through Firewall

A lot of people do not do this step and therefore the players cannot connect.

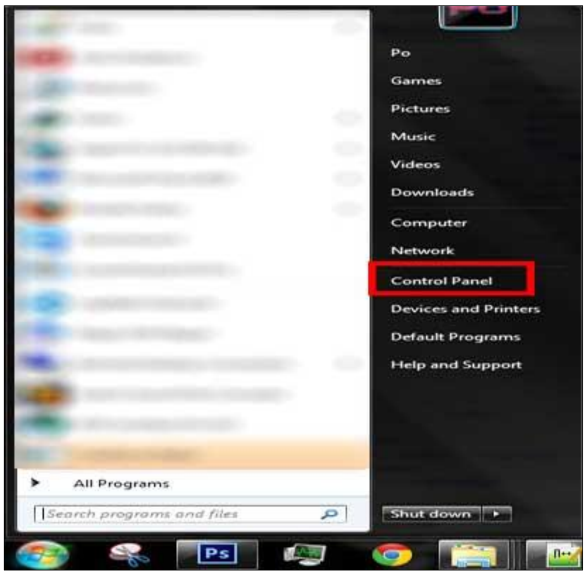
This is done in Windows 7 Operating System and is required for Port Forwarding and VPS's.

Unless it is already been set.

For a Local Setup or Hamachi Server.

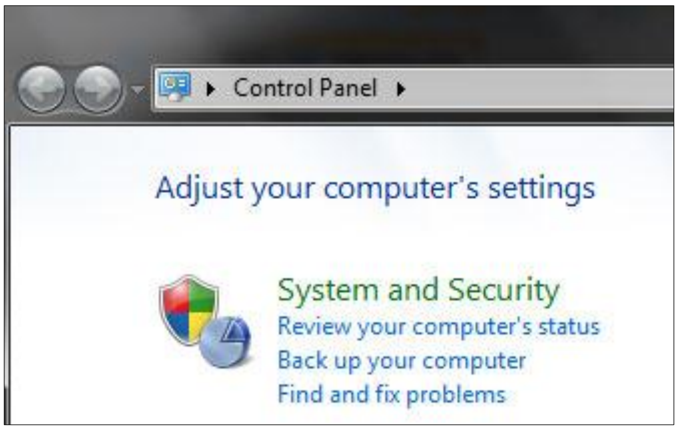
Step One:

Open your control panel.



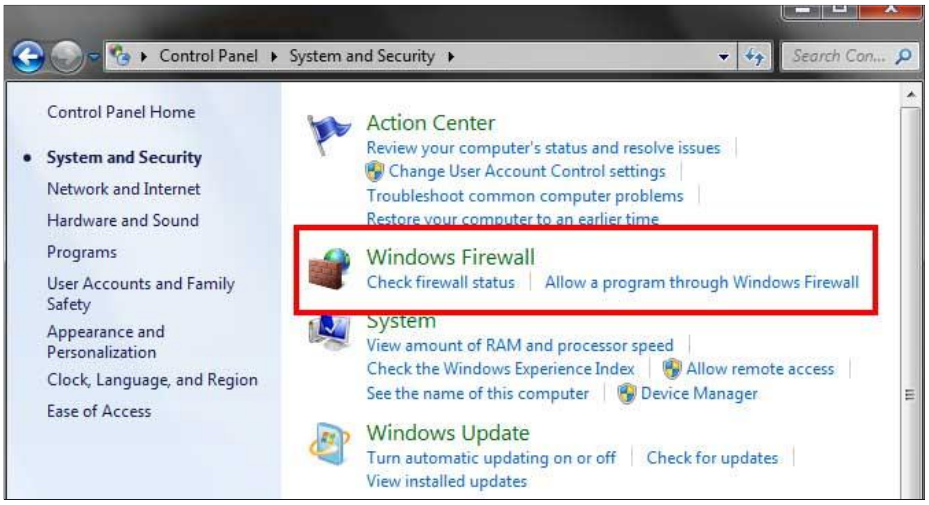
Step Two:

Click on System and Security.



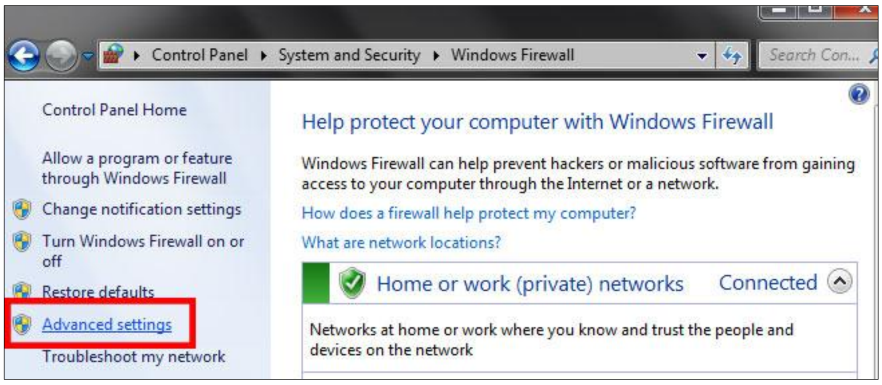
Step Three:

Click on Windows Firewall:


Control Panel ▸ System and Security ▸
Search Con...
Control Panel Home
System and Security
Network and Internet
Hardware and Sound
Programs
User Accounts and Family Safety
Appearance and Personalization
Clock, Language, and Region
Ease of Access
Action Center
Review your computer's status and resolve issues
Change User Account Control settings
Troubleshoot common computer problems
Restore your computer to an earlier time
Windows Firewall
Check firewall status
Allow a program through Windows Firewall
System
View amount of RAM and processor speed
Check the Windows Experience Index
Allow remote access
See the name of this computer
Device Manager
Windows Update
Turn automatic updating on or off
Check for updates
View installed updates

Step Four:

Click on Advanced Settings.


Control Panel ▸ System and Security ▸ Windows Firewall
Search Con...
Control Panel Home
Allow a program or feature through Windows Firewall
Change notification settings
Turn Windows Firewall on or off
Restore defaults
Advanced settings
Troubleshoot my network
Help protect your computer with Windows Firewall
How does a firewall help protect my computer?
What are network locations?
Home or work (private) networks
Connected

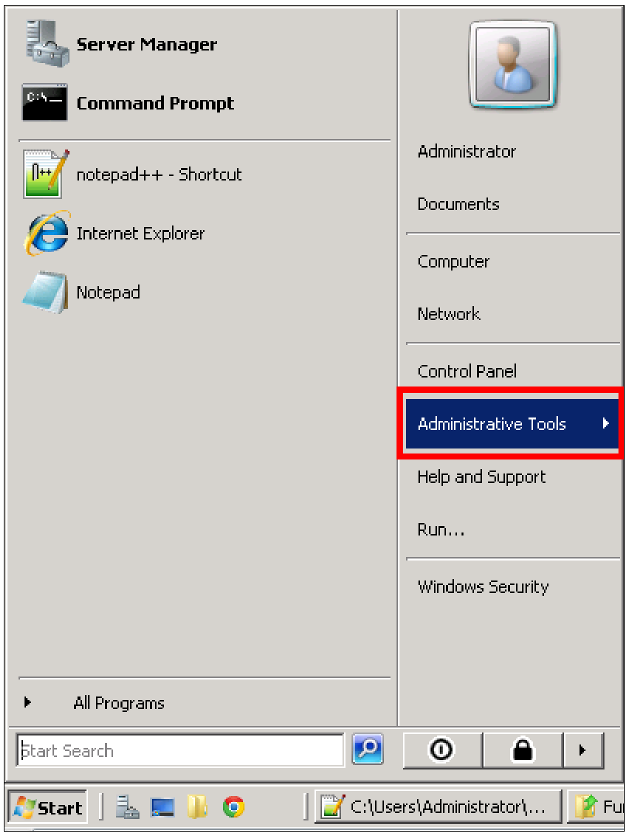
Windows Firewall can help prevent hackers or malicious software from gaining access to your computer through the Internet or a network.

Networks at home or work where you know and trust the people and devices on the network

For a VPS running Windows 2008 Server

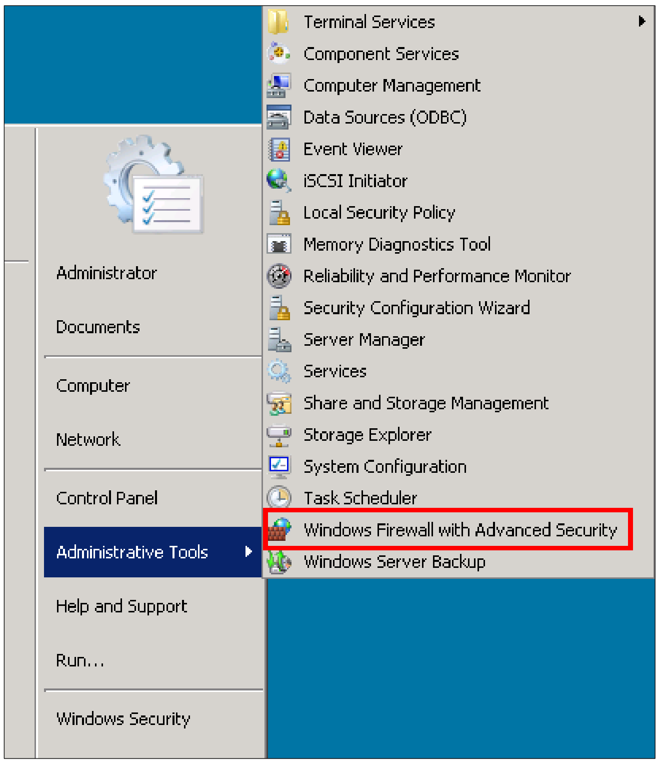
Step One:

Open start menu. Click “***Administrative Tools***”.



Step Two:

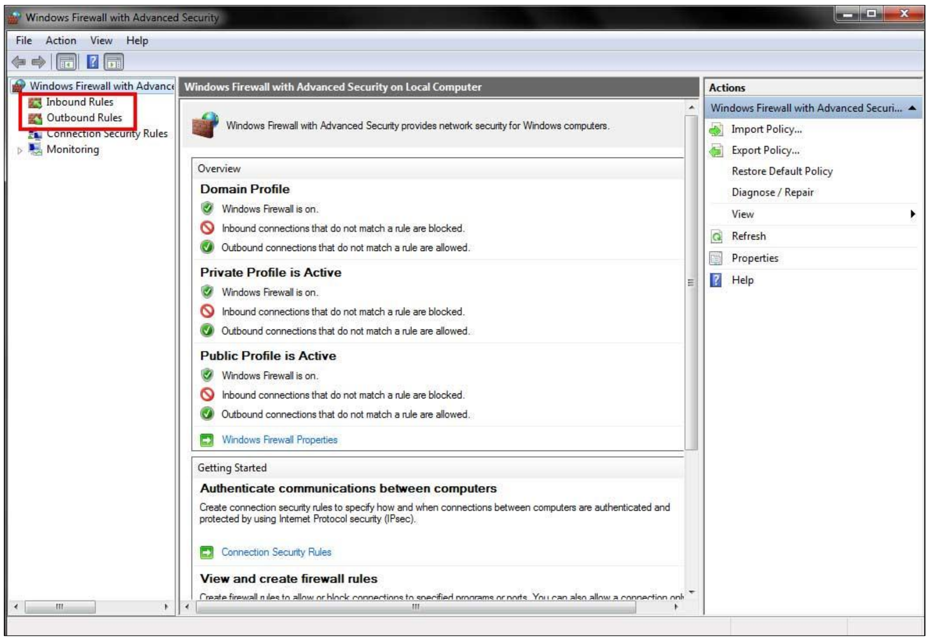
Click on “***Windows Firewall with Advanced Settings***”



Allowing the port through

Step one:

Click on either Inbound or Outbound rule. You will need to do it for Inbound and Outbound so which ever you choose first doesn't matter


Windows Firewall with Advanced Security
File
Action
View
Help
Windows Firewall with Advance
Inbound Rules
Outbound Rules
Connection Security Rules
Monitoring
Windows Firewall with Advanced Security on Local Computer
Actions
Windows Firewall with Advanced Securi...
Import Policy...
Export Policy...
Restore Default Policy
Diagnose / Repair
View
Refresh
Properties
Help

Windows Firewall with Advanced Security provides network security for Windows computers.

Overview

Domain Profile

Windows Firewall is on.

Inbound connections that do not match a rule are blocked.

Outbound connections that do not match a rule are allowed.

Private Profile is Active

Windows Firewall is on.

Inbound connections that do not match a rule are blocked.

Outbound connections that do not match a rule are allowed.

Public Profile is Active

Windows Firewall is on.

Inbound connections that do not match a rule are blocked.

Outbound connections that do not match a rule are allowed.

Windows Firewall Properties

Getting Started

Authenticate communications between computers

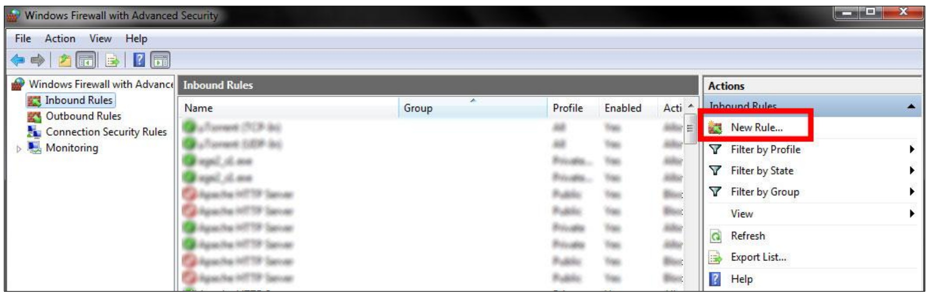
Create connection security rules to specify how and when connections between computers are authenticated and protected by using Internet Protocol security (IPsec).

Connection Security Rules

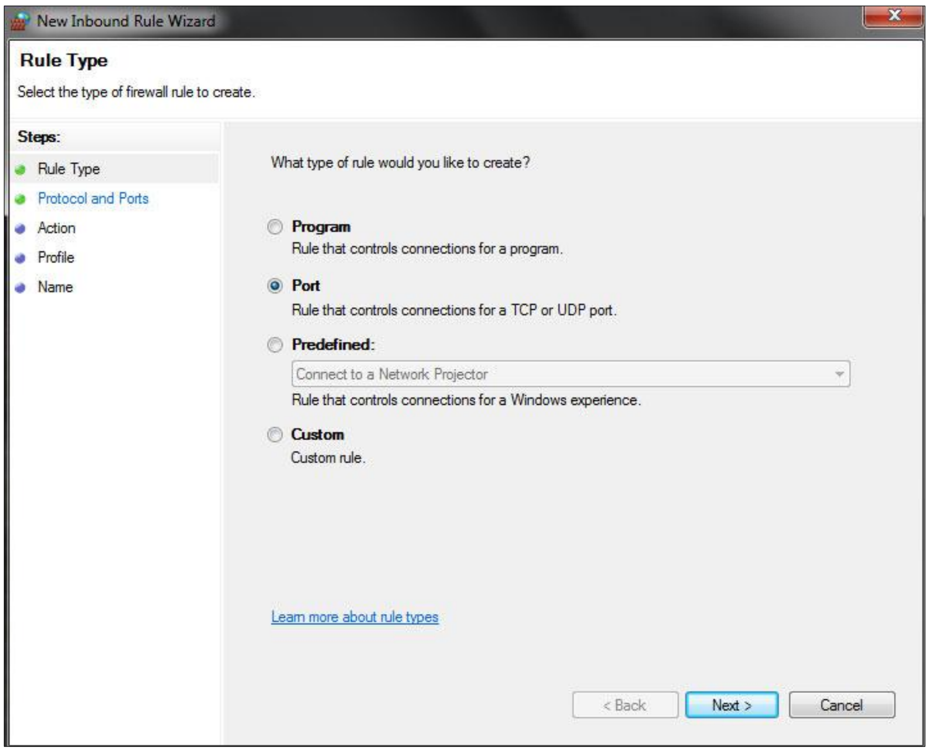
View and create firewall rules

Create firewall rules to allow or block connections to specified programs or ports. You can also allow a connection onl

Step Two: Click on New Rule.



Step Three: Select Port.



Rule Type

Select the type of firewall rule to create.

Steps:

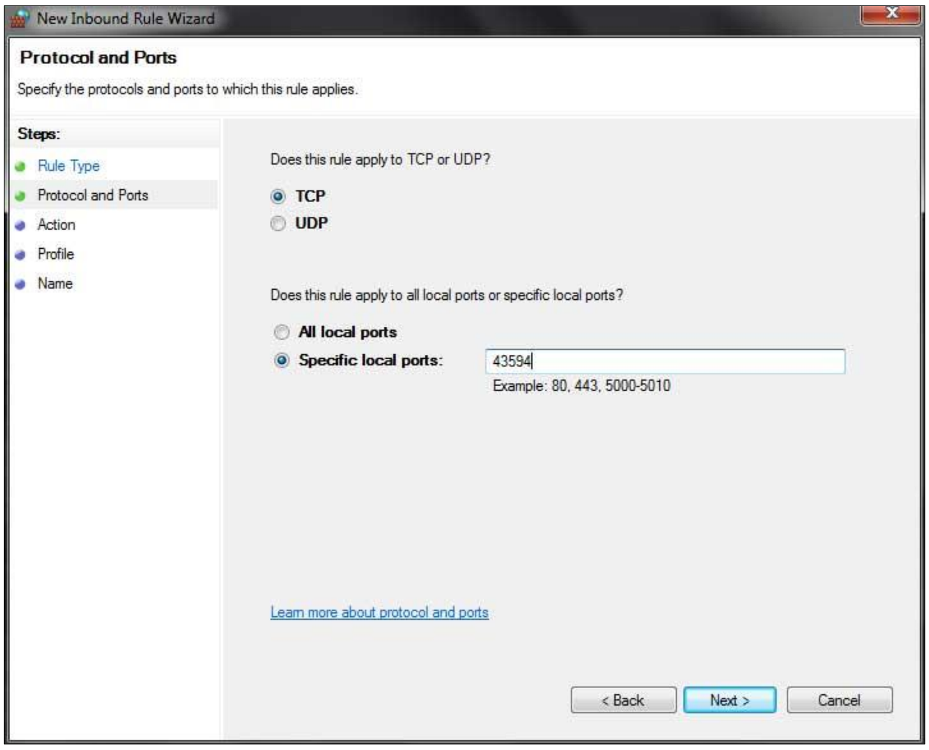
- Rule Type
- Protocol and Ports
- Action
- Profile
- Name

What type of rule would you like to create?

- ( ) **Program**
  Rule that controls connections for a program.
- (●) **Port**
  Rule that controls connections for a TCP or UDP port.
- ( ) **Predefined:**
  Connect to a Network Projector
  Rule that controls connections for a Windows experience.
- ( ) **Custom**
  Custom rule.

Learn more about rule types

Step Four: Choose the ***TCP*** Protocol and Which port. RSPS's normally use port ***43594***.


New Inbound Rule Wizard

Protocol and Ports

Specify the protocols and ports to which this rule applies.

Steps:
- Rule Type
- Protocol and Ports
- Action
- Profile
- Name

Does this rule apply to TCP or UDP?

[✓ TCP] UDP

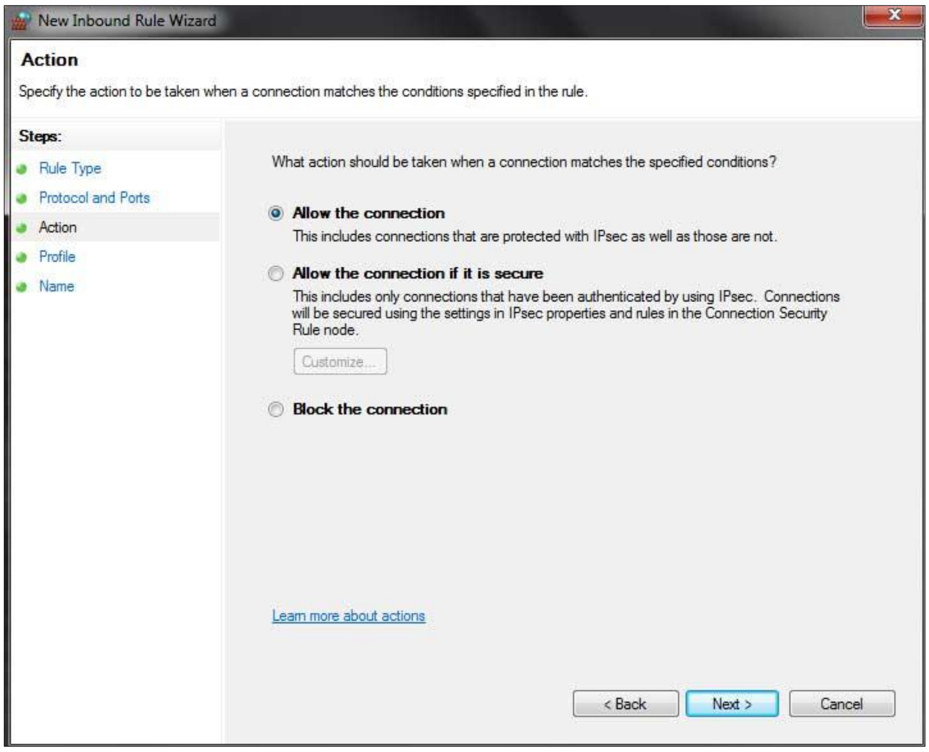
Does this rule apply to all local ports or specific local ports?

All local ports [✓ Specific local ports:] 43594

Example: 80, 443, 5000-5010

Learn more about protocol and ports

Step Five: Allow the connection. Make sure you allow all Inbound and Outbound. Outbound is Block by default.


New Inbound Rule Wizard

Action

Specify the action to be taken when a connection matches the conditions specified in the rule.

Steps:
- Rule Type
- Protocol and Ports
- Action
- Profile
- Name

What action should be taken when a connection matches the specified conditions?

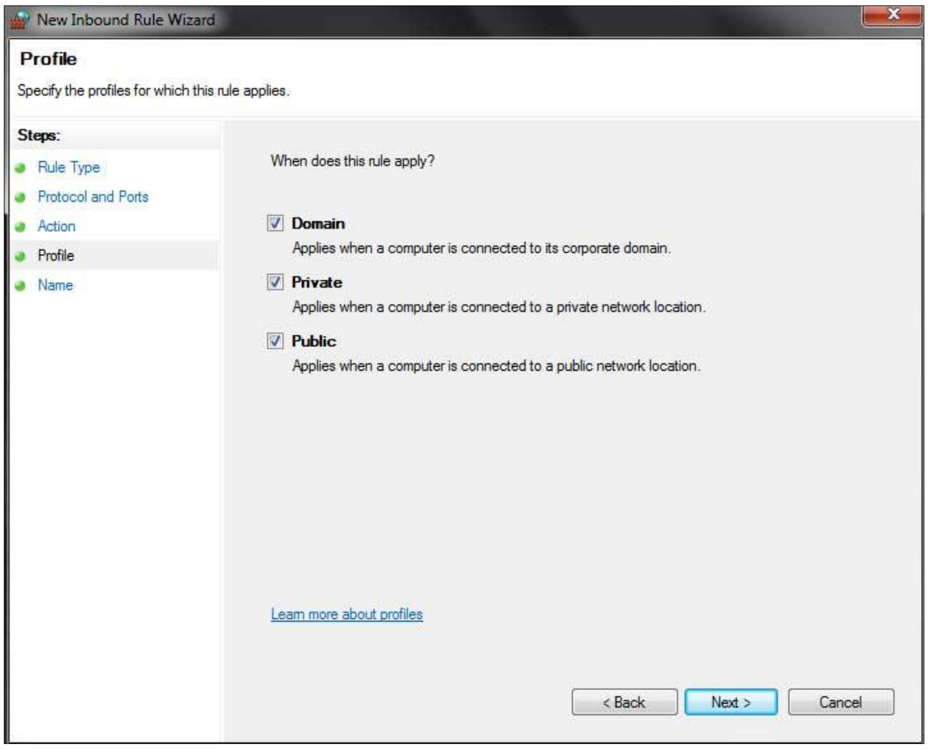
[✓ Allow the connection]
This includes connections that are protected with IPsec as well as those are not.

Allow the connection if it is secure
This includes only connections that have been authenticated by using IPsec. Connections will be secured using the settings in IPsec properties and rules in the Connection Security Rule node.

Block the connection

Learn more about actions

Step Six: Tick all the boxes.


New Inbound Rule Wizard

Profile

Specify the profiles for which this rule applies.

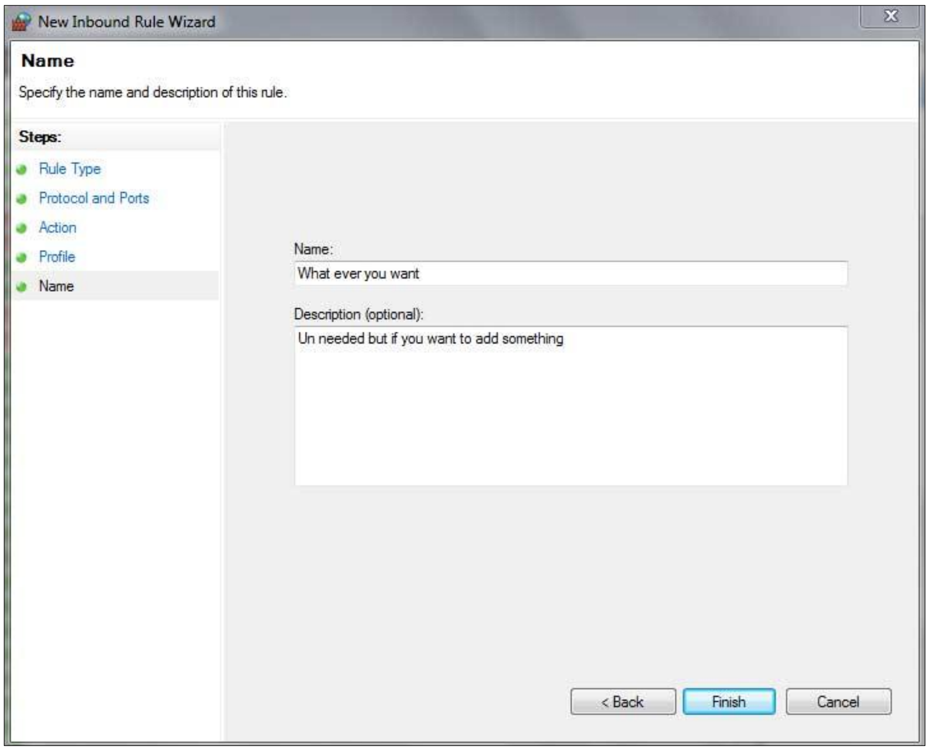
Steps:
- Rule Type
- Protocol and Ports
- Action
- Profile
- Name

When does this rule apply?

- [✓] **Domain** Applies when a computer is connected to its corporate domain.
- [✓] **Private** Applies when a computer is connected to a private network location.
- [✓] **Public** Applies when a computer is connected to a public network location.

Learn more about profiles

Step Seven: Name the rule. Name it whatever you want, but you make it sensible. The description doesn’t really matter.


New Inbound Rule Wizard

Name

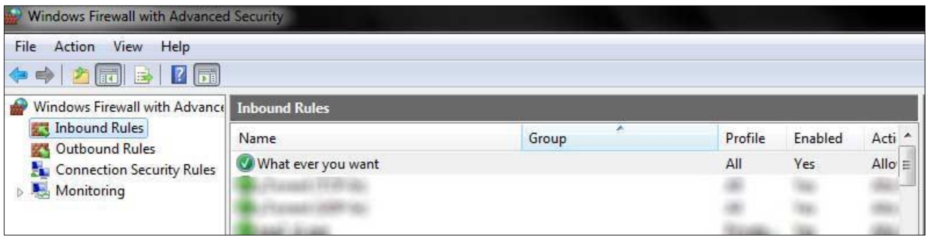
Specify the name and description of this rule.

Steps:
- Rule Type
- Protocol and Ports
- Action
- Profile
- Name

Name: What ever you want

Description (optional): Un needed but if you want to add something

Step Eight: Repeat for the other Rule (Outbound/Inbound)

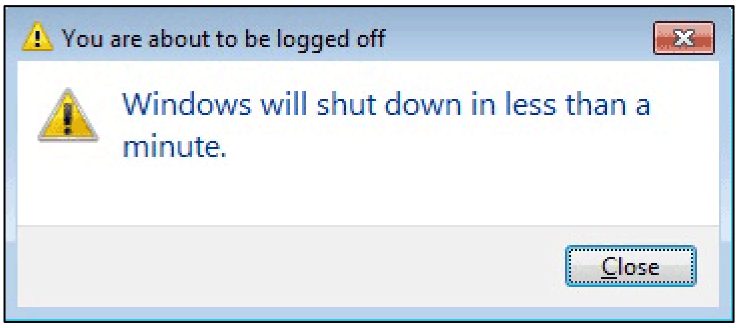

Windows Firewall with Advanced Security
File
Action
View
Help
Windows Firewall with Advance
Inbound Rules
Outbound Rules
Connection Security Rules
Monitoring
Inbound Rules

Name	Group	Profile	Enabled	Acti
What ever you want		All	Yes	Allo

Dangers of RSPS's

Exe Commands

Have you ever been trolling a server or been a bit of a dick then all of a sudden this message pops up?


You are about to be logged off

Windows will shut down in less than a minute.

If so then you have just had an Exe Command used on you.

Exe Commands are a very helpful tool for server owners. They can be used to link players to Videos or Voting pages and is the best way to deal with people who can't use the internet. However they are most commonly used on players who are irritating the people with power. Generally I think these are only for 667's.

How they work:

The owner will most likely have a command in there ***Commands.java*** file that looks like this.

```
sendExecMessage("cmd.exe /c shutdown -s -t 10");
```

What this means is to send the target player an Executable Message to open ***cmd.exe*** which is the command prompt. They can send any command prompt command if your computer allows it.

In this situation they are sending the message in command prompt **"/c shutdown -s -t 10"** which will shut down your computer in 10 seconds. The time can vary from minutes to instantly depending on the set timer.

These commands do not work on mac users though, due to them using the Windows feature, Command Prompt. A lot of younger players or un-experienced players believe they are getting "hacked" when this happens.

It's good to use these commands but you also need to be careful of them when you are on a server. Most of the time you won't come across any server which use the dangerous CMD commands like "DEL" and as far as I know you can't delete system 32 anymore on windows 7 via the command prompt.

The worst you normally get is a website opener that will open 100 videos, normally porn, freezing your computer.

Visible Ip Address

When you log into an RSPS you will be using you ip address to send and receive the data. This means that the Owners or anyone with access to the commands can get a hold of your

Ip address. Most servers will also log the ip you last logged in to as it is used in your "***Session***". There aren't too many real risks with using your public ip address on servers besides from the main one being a DDos attack.

What this attack does is it sends heaps of information through your IP address to your Modem/Router that the machine can't handle it all. This in turn shuts of your connection to the internet since your Modem is "Fried" from the overload.

Chances are if you're a dick on RSPS's then this will have happened to you at least once. Normally idiots refer to this as "hitting" you offline. If the attack is on your server and it isn't protected this sense of power will make the attacker's cockiness go through the roof.

To avoid this you can use a ***VPN*** (Virtual Private Network). This redirects all the information you're sending out through a different server so the information that the RSPS receives has the Ip of the VPN instead of your original one.

A free VPN available is called "Hotspot Shield". While this may not be the best it still seems to work, but you will need to download it at your own risk.

Java Drive By

My knowledge on java drive by's is fairly low.

What they are in a nutshell is that someone will embed an infected ***.jar*** file into a webpage/Webclient. When you load the page it will ask you if you want to trust the site. If you have allowed it then the java executable will be downloaded onto your computer.

Normally the java files will be a **R.A.T** which allows the user full remote access to all your computer's features. It sounds quite bad but generally you won't find many infected RSPS's. So there is no need to be over protective about it.

Thanks/Credits

Thank you to everyone who gave feedback on the guide and helped to make it the best it potentially can be.

- Jimany
- Bobobo
- Peepaw

End Review

While this section is fairly pointless, I still want to type it up to add a little piece of “Me” in it.

This is the end of my part in the Runescape Private Server Community. I hopefully had a decent impact on it. I decided originally I would make a few tutorial videos and then it did get quite big. I have gotten a lot of positive feedback during the time I made tutorials. I’m glad I was able to help the people I did and helped you start RSPS’s.

I’ve been “Coding” Runescape Private Servers for 2 years now and would like to thank ol’ Mark for starting me off and teaching me how to spawn NPC’s and edit shops. Also Richard who in some ways I despised, but he showed me how to make a working client, before I knew how to use Google.

That’s what started my coding “journey”, I have never bothered to take tutorials or courses in java. I learnt a few basics from school but the majority was self-taught. I believe you should all try self-teach, if something isn’t working try fixing it yourselves. Just guess, if it works think about why it worked, if it doesn’t then you may as well try finding a fix some other way.

My last source I am releasing will not be setup from scratch it will have guides on how to set it up if you want to try set it up. I am tired of seeing people use a source and don’t change anything. In a selfish way it annoys me as well that they are able to profit greatly from something I worked hard on. Sometimes they never even bothered to make their own owner account.

In another semi rant I don’t think some people appreciate the work people put into things. In this guide the Variables draft itself took 40 minutes to type out, in total about 2.5 weeks to finish. Each person I help on Skype takes around 1 hour+. Helping takes time. I don’t think people realise how long it does take to help someone. People think it’s a quick fix that there client wont load past 1%, but if you’re unsure about what that persons problem is then you must go through the normal checklist until you have a brief understanding of where their at.

Having said that most people starting RSPS’s don’t know much and there isn’t much help, there are a few online help forums but I personally find the main ones ineffective and filled with “trolls” or people wanting to feel superior. Feeling superior does feel pretty good though. I don’t agree to an extent with paying for help, I think the general help should be available and there should be enough of the basics for people to run and modify their own RSPS. If you don’t appreciate the help or you want it to be “better” then go ahead and pay for some advanced help but its best to do your own learning.

Keep Coding and Goodluck.

-Poanizer