

Forum:[Hack Forums](#)

Thread: [Java Multi Platform Save Directory](#)

+Haze - June 24, 2015, 1:10 pm

Okay so I'm currently creating a game and I've having a problem with the save feature. Currently I use WindowsOS and the Windows side of the code works a treat, but since I don't have other OS systems like mac or linux I don't know if this code would work on all platforms (the major ones)

Here the code.

Spoiler ([Click to View](#))

Code private Path getSavePath() {

```
    try {
```

```
        String OS = System.getProperty("os.name").toLowerCase();
```

```
        Path fullDirectory = null;
```

```
        Path fullPath = null;
```

```
        if (OS.contains("win")) {
```

```
            fullDirectory = Paths.get(System.getenv("AppData") + "\\NoughtPlusPlus");
```

```
            fullPath = Paths.get(fullDirectory.toString() + "\\RecentScores.npp");
```

```
        } else {
```

```
            fullDirectory = Paths.get(System.getProperty("user.home") + "\\Library\\Application  
Support\\NoughtPlusPlus");
```

```
            fullPath = Paths.get(fullDirectory.toString() + "\\Library\\Application  
Support\\RecentScores.npp");
```

```
        }
```

```
        if (Files.notExists(fullDirectory))
```

```
            Files.createDirectory(fullDirectory);
```

```
        return fullPath;
```

```
    } catch (IOException e) {
```

```
        MessageBox.show("Error trying to read saved game data - Press okay to see error");
```

```
        MessageBox.show(e.getMessage());
```

```
        return null;
```

```
    }
```

```
}
```

So, what do people suggest or what is the right way for obtaining like an AppData folder for other platforms like Mac and Linux?

I would also appreciate replies from Mac & Linux users too

Thanks

JavaProfit - June 24, 2015, 1:40 pm

For mac use ~/Library/Application Support

... Not sure what people usually use for most distros of Linux, but somewhere in the home folder works.

ninja: I don't think you can use backslashes for anything other than dos. File.separator or just "/" works.

+Haze - June 24, 2015, 4:18 pm

So, for a mac and linux user.

Will

Code fullDirectory = Paths.get(System.getProperty("user.home") + "\\Library\\Application Support\\NoughtPlusPlus");

fullPath = Paths.get(fullDirectory.toString() + "\\Library\\Application Support\\RecentScores.npp");
Work?

Also, seems you're right about the backslash thing, thanks :D

JershDervis - June 25, 2015, 7:42 pm

I know the Minecraft Launcher uses the following method:

Code public static final String APPLICATION_NAME = "minecraft";

```
public enum OS
{
    WINDOWS, MACOS, SOLARIS, LINUX, UNKNOWN;
}

public static OS getPlatform()
{
    String osName = System.getProperty("os.name").toLowerCase();
    if (osName.contains("win")) {
        return OS.WINDOWS;
    }
    if (osName.contains("mac")) {
        return OS.MACOS;
    }
    if (osName.contains("linux")) {
        return OS.LINUX;
    }
    if (osName.contains("unix")) {
        return OS.LINUX;
    }
    return OS.UNKNOWN;
}
```

```
public static File getWorkingDirectory()
```

```

{
    String userHome = System.getProperty("user.home", ".");
    File workingDirectory;
    switch (getPlatform().ordinal())
    {
    case 1:
    case 2:
        workingDirectory = new File(userHome, "." + APPLICATION_NAME + "/");
        break;
    case 3:
        String applicationData = System.getenv("APPDATA");
        String folder = applicationData != null ? applicationData : userHome;

        workingDirectory = new File(folder, "." + APPLICATION_NAME + "/");
        break;
    case 4:
        workingDirectory = new File(userHome, "Library/Application Support/" +
APPLICATION_NAME);
        break;
    default:
        workingDirectory = new File(userHome, APPLICATION_NAME + "/");
    }
    return workingDirectory;
}

```

JavaProfit - June 25, 2015, 8:22 pm

(Jun 25, 2015 07:42 PM)JershDervis Wrote: [\[CITE\]](#) I know the Minecraft Launcher uses the following method:

Code public static final String APPLICATION_NAME = "minecraft";

```

public enum OS
{
    WINDOWS, MACOS, SOLARIS, LINUX, UNKNOWN;
}

public static OS getPlatform()
{
    String osName = System.getProperty("os.name").toLowerCase();
    if (osName.contains("win")) {
        return OS.WINDOWS;
    }
    if (osName.contains("mac")) {
        return OS.MACOS;
    }
    if (osName.contains("linux")) {
        return OS.LINUX;
    }
    if (osName.contains("unix")) {
        return OS.LINUX;
    }
    return OS.UNKNOWN;
}

```

```

public static File getWorkingDirectory()
{
    String userHome = System.getProperty("user.home", ".");
    File workingDirectory;
    switch (getPlatform().ordinal())
    {
        case 1:
        case 2:
            workingDirectory = new File(userHome, "." + APPLICATION_NAME + "/" );
            break;
        case 3:
            String applicationData = System.getenv("APPDATA");
            String folder = applicationData != null ? applicationData : userHome;

            workingDirectory = new File(folder, "." + APPLICATION_NAME + "/" );
            break;
        case 4:
            workingDirectory = new File(userHome, "Library/Application Support/" +
APPLICATION_NAME);
            break;
        default:
            workingDirectory = new File(userHome, APPLICATION_NAME + "/" );
    }
    return workingDirectory;
}

```

That's decompiled from the launcher, right? I'm just assuming that because enum switches use ordinal() after being compiled and no one writes enum switches with ordinal

DragonBook - June 26, 2015, 4:51 am

If you truly want it to be multi-platform, save yourself the trouble and 1) allow the user to choose their own directory and 2) store all save profiles and settings files in a sub-directory. Don't worry about how Mojang does it. You can always make it more sophisticated later.

+Haze - June 26, 2015, 4:38 pm

(Jun 26, 2015 04:51 AM)___isoc99_scanf Wrote: [\[CITE\]](#) If you truly want it to be multi-platform, save yourself the trouble and 1) allow the user to choose their own directory and 2) store all save profiles and settings files in a sub-directory. Don't worry about how Mojang does it. You can always make it more sophisticated later.

What sub directory should I save it to?