

# **WEJAC**

## **UserBook**



**Will's Elided Java Api Compiler**

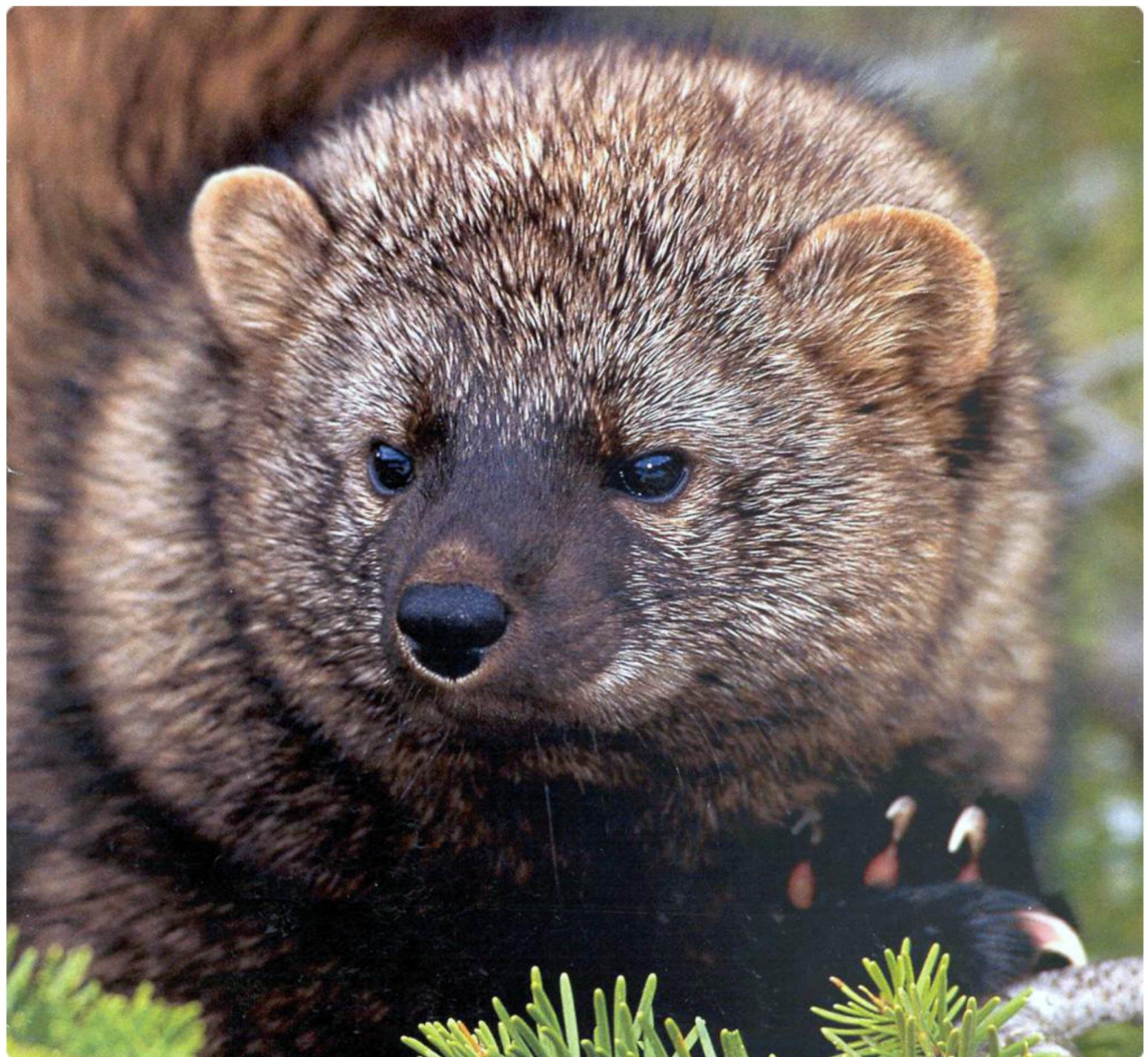
**William F. Gilreath**

William Gilreath

# WEJAC

## Will's Elided Java Api Compiler

July 2019



© Copyright 2019 by William F. Gilreath. All Rights Reserved.

Licensed under the Creative Commons Attribution-ShareAlike 4.0 International  
(CC BY-SA 4.0)

# Table of Contents

Forward	5
Chapter 1. Compiler Installation	6
<i>Pre-Installation Steps</i>	6
<i>Installation Steps</i>	8
<i>Post-Installation Steps</i>	9
Chapter 2. Compiler Parameters	11
<i>Kinds of Parameters</i>	11
<i>Compilation</i>	11
<i>Error Reporting</i>	11
<i>Help or Information</i>	12
<i>javac Parameters</i>	12
Chapter 3. Compiler Options	13
<i>brief</i>	13
<i>echo</i>	13
<i>final</i>	14
<i>help</i>	14
<i>hush</i>	15
<i>info</i>	15
<i>javac</i>	16
<i>mute</i>	16
<i>time</i>	17
<i>Multiple Compiler Options</i>	17
Afterword	18
<i>Colophon</i>	18
<i>Author</i>	18
<i>Contributing</i>	18
License	19

# Forward

Why another Java compiler? Simple—for simplicity in use of options and consistency in error reporting.

WEJAC is an acronym for Will's Elided Java Api Compiler. The name explains the compiler purpose; the primary word is 'elided' from the word elision, which means "to omit or to leave out of consideration." WEJAC omits the dizzying number of options for the underlying Java compiler accessed through the Java Compiler API as JSR 199. The number of options to use WEJAC to compile Java source code is considerably simpler than the javac compiler.

WEJAC also is consistent in error reporting. Some errors such as deprecation with the javac compiler require re-compilation—yet another compiler option. WEJAC always provides an error report, the line and position, and the source code fragment to illustrate where the cause of the error is in the overall Java source code. Thus error message, location, and illustrative cause.

The javac compiler has many options and features—and WEJAC provides the option to use other javac compiler options. But in general, I have found a simpler option set for compiler parameters best. WEJAC is an example of a software development engineer wanting a simpler tool, so then implementing, deploying, sharing, and here now—documenting it.

Happy Compiling!

William Gilreath

July 2019

# Chapter 1. Compiler Installation

WEJAC is written in Java, and compiled into a .jar file. Thus the compiler is “bootstrapped” in the programming language it compiles using the Java Compiler API. Thus any platform (Windows/Linux/macOS) can download, install, configure, and use WEJAC.

WEJAC consists of the binary bytecode executable file as a JAR file (a Java Archive file) and a wrapper “kicker” script or batch file that executes the JAR file on the Java Virtual Machine (JVM), passing the command-line arguments to the Java Virtual Machine.

## Pre-Installation Steps

### 1. Checking for the OpenJDK

In order to use the WEJAC compiler, first the system used must have the OpenJDK installed. There are many OpenJDK downloads from different vendors. The simplest way is to open a console or terminal window, and enter the following command:

```
$java -version
```

If the OpenJDK and Java environment are installed the result is:

```
$java -version
java version "12.0.1" 2019-04-16
Java(TM) SE Runtime Environment (build 12.0.1+12)
Java HotSpot(TM) 64-Bit Server VM (build 12.0.1+12, mixed mode,
sharing)
```

If not, download and install the OpenJDK on your system.

The WEJAC compiler runs on the JDK 7 and upwards to (currently at the time of this writing) JDK 12. Thus it is vital to install a compatible OpenJDK Java environment. Check the Java release version is compatible, greater than or equal to Java release version 1.7.x.

## 2. Download the WEJAC package

The entire WEJAC package is downloadable from Will's GitHub repo. The repo is at the link:

<https://github.com/wgilreath/WEJAC>

The ZIP archive is at the specific link of:

[https://github.com/wgilreath/WEJAC/WEJAC\\_July\\_2019.zip](https://github.com/wgilreath/WEJAC/WEJAC_July_2019.zip)

## 3. Unzip the WEJAC package.

Use your favorite ZIP utility to unzip and decompress the WEJAC package into a directory.

# Installation Steps

The installation is simply install the wejac shell script, the WEJAC.jar binary file, and on macOS/Linux systems the wejac.1 man or manual page.

1. Install wejac shell script.

On my MacBook, the wejac shell script is installed in the directory: /usr/local/bin/

In a terminal window wejac is installed by the shell commands:

```
$sudo bash  
Password:  
#cp Downloads/wejac /usr/local/bin
```

2. Install the WEJAC.jar JAR file.

On my MacBook, the WEJAC.jar file is in the directory: /usr/local/lib/

In a terminal window WEJAC.jar is installed by the shell commands:

```
$sudo bash  
Password:  
#cp Downloads/WEJAC.jar /usr/local/lib
```

3. For Linux/macOS systems, install the manual page.

On my MacBook the wejac.1 man page is in the directory: /usr/local/share/man/man1/

In a terminal window wejac.1 is installed by the shell commands:

```
$sudo bash  
Password:  
#cp Downloads/wejac.1 /usr/local/share/man/man1
```

## Post-Installation Steps

1. Verify that the installation successful.

Use the command wejac in a terminal or console window; if successfully installed, the result will be:

```
$wejac  
WEJAC - Will's Elided Java Api Compiler  
(C) Copyright 2019 William F. Gilreath. All Rights Reserved Version 1.3  
Released July 2019  
License is GNU General Public License (GPL) version 3.0  
  
Error! No compiler options or files given! Use -help for options.  
$
```

2. Compile simple if ubiquitous HelloWorld.java source code file. In your favorite text editor write the simple Java source:

```
public final class HelloWorld{  
  
    public final static void main(String[] args){  
        System.out.println("Hello, World!!!");  
        System.exit(0);  
    }  
}
```

In a terminal or console window, compile HelloWorld.java using the WEJAC compiler:

```
$wejac HelloWorld.java  
$
```

Use the -echo option to confirm the WEJAC compiler successfully compiled the HelloWorld.java source code:

```
$wejac -echo HelloWorld.java  
  
Java Compile Options: [-g]  
  
Compiler result for file: 'HelloWorld.java' is: Success.  
$
```

### 3. Contribute, and share your experience with WEJAC.

WEJAC is open-source so that motivated, bright, and curious software developers can contribute to the project. Also share your experience with the WEJAC compiler with other developers.

## Chapter 2. Compiler Parameters

WEJAC is designed to have fewer command-line options to use with the compilation of Java source files. These options are organized into different kinds of options used with the WEJAC tool. Overall there are only **nine** options for the WEJAC compiler.

### Kinds of Parameters

WEJAC has four kinds of options:

1. Compilation
2. Error Reporting
3. Help or Information
4. javac

### Compilation

The options for compilation are for the compilation of Java source files. The compiler options are:

1. -echo - Print Java compiler options and success or failure.
2. -final - Compile final release without debug information.
3. -time - Print total time for success compiling of a source file.

### Error Reporting

The options for error reporting when compiling Java source files. The error reporting options are:

1. -brief - Print only a brief count of compiler messages.
2. -hush - Disable all compiler messages except errors.
3. -mute - Disable all compiler messages.

## **Help or Information**

The options for help listing the options or information about the WEJAC Java compiler. There are two options but with different parameters:

1. -help – Print list of compiler options and exit.
2. -info – Print compiler version information and exit.

## **javac Parameters**

The javac option specifies the following parameter is a standard Java compiler option that is passed to the WEJAC compiler as-is. This allows the flexibility of using a javac compiler option without having to check or add the javac option to WEJAC.

# Chapter 3. Compiler Options

This chapter contains a list of all WEJAC compiler options with a short example to illustrate with the compiler option does in practice.

## brief

The -brief compiler option specifies a brief, or a summary of the number of error diagnostics.

```
wejac -brief HelloWorld.java
 1 Diagnostic messages:
   1 Error
```

## echo

The -echo compiler option specifies the compiler “echo” the javac compiler parameters passed, and the final result of compiling a Java source file.

When HelloWorld.java is compiled using the WEJAC compiler with the -echo option:

```
wejac -echo HelloWorld.java

Java Compile Options: [-g]

Compiler result for file: 'HelloWorld.java' is: Success.
```

When HelloWorldBad.java is compiled using the WEJAC compiler with the -echo option:

```
$wejac -echo HelloWorldBad.java

Java Compile Options: [-g]

Error: HelloWorldBad.java.
Line 4 At 24: unclosed string literal
  System.out.println("Hello, World!!!!");
                           ^
Compiler result for file: 'HelloWorldBad.java' is: Failure!
```

## **final**

The -final compiler option specifies a non-debug, final release compile of the Java source file. Using the -echo compiler option, the javac compiler parameter is shown.

When HelloWorld.java is compiled using the WEJAC compiler with the -final option:

```
$wejac -echo -final HelloWorld.java  
Java Compile Options: [-g:none]  
Compiler result for file: 'HelloWorld.java' is: Success.
```

## **help**

The compiler option -help prints the helpful information of the various command-line options to the WEJAC Java compiler, and then exits without compiling.

```
$wejac -help  
Usage: wejac (option)* [ -javac (javac-options)+ ] (java-file)+ | ( -help | -info )  
WEJAC OPTIONS:  
Compiler Options: [ -echo ] | [ -final ] | [ -time ]  
-echo      Print Java compiler options and success or failure.  
-final     Compile final release without debug information.  
-time      Print total time for success compiling of a source file.  
Error Reporting Option: [ -brief | -hush | -mute ]  
-brief     Print only a brief count of compiler messages.  
-hush      Disable all compiler messages except errors.  
-mute      Disable all compiler messages.  
Help or Version Option: ( -help | -? ) | ( -info | -v )  
-help      Print list of compiler options and exit.  
-info      Print compiler version information and exit.  
Note: All options for -javac are passed as-is to the compiler.
```

## **hush**

The compiler option -hush will “hush” all compiler diagnostics, except for compiler errors. Other compiler diagnostics such as warnings, notes, and mandatory warnings are suppressed.

When Unchecked.java is compiled with the WEJAC compiler, it generates a deprecation warning:

```
$wejac Unchecked.java
Error: Unchecked.java.
Unchecked.java uses or overrides a deprecated API.

Error: Unchecked.java.
Recompile with -Xlint:deprecation for details.
```

When the -hush options is used, and Unchecked.java is compiled with the WEJAC compiler, there is no warning:

```
$wejac -hush Unchecked.java
$
```

## **info**

The compiler option -info gives the information about the compiler—such as the release, version, copyright, and software license.

```
$wejac -info
WEJAC - Will's Elided Java Api Compiler
(C) Copyright 2019 William F. Gilreath. All Rights Reserved
Version 1.3 Released July 2019
```

## **javac**

The compiler option -javac allows a Java compiler option to be passed to the underlying Java compiler.

When Unchecked.java is compiled with the WEJAC compiler, it generates a deprecation warning:

```
$wejac Unchecked.java
Error: Unchecked.java.
Unchecked.java uses or overrides a deprecated API.

Error: Unchecked.java.
Recompile with -Xlint:deprecation for details.
```

When Unchecked.java is compiled with the WEJAC compiler and the -javac option of -Xlint:deprecation is used the details are given of the deprecation warning:

```
$wejac -javac -Xlint:deprecation Unchecked.java
Error: Unchecked.java.
Line 29 At 18: [deprecation] Date(int,int,int) in java.util.Date has
been deprecated
    Date d = new Date(86, 04, 05);           // May 5, 1986
                                              ^
```

## **mute**

The compiler option -mute suppresses or “mutes” all compiler diagnostic messages from the WEJAC compiler. When HelloWorldBad.java is compiled with the -mute option:

```
$wejac -mute HelloWorldBad.java
$
```

## time

The compiler option -time gives the total time in milliseconds to compile a Java source file with the WEJAC compiler. When HelloWorld.java is compiled with the -time option:

```
$wejac -time HelloWorld.java  
Time: 409-ms for: HelloWorld.java  
$
```

## Multiple Compiler Options

The WEJAC compiler allows the mixing of compiler options, but using only a single error reporting option. Consider compiling HelloWorld.java with the WEJAC compiler using the -echo -time options:

```
$wejac -echo -time HelloWorld.java  
  
Java Compile Options: [-g]  
  
Time: 411-ms for: HelloWorld.java  
Compiler result for file: 'HelloWorld.java' is: Success.  
$
```

## **Afterword**

The author and creator of WEJAC welcomes comments, questions, bug reports, and thoughts from users. Please feel free to e-mail ([wfgilreath@yahoo.com](mailto:wfgilreath@yahoo.com)) with the subject "WEJAC" to connect.

## **Colophon**

The cute animal on the cover is a "Wejack" the Wejack (or more commonly referred to as a Fisher), is a small mammal native to North America.

## **Author**

The WEJAC compiler is created and developed by Will or William F. Gilreath. Will is a software development engineer, computer scientist, mathematician, writer, and poet. Will has been writing code starting when he was a boy with a Sinclair ZX-81, and then moving on to the Commodore 64 C-BASIC, and later GW-BASIC on a Tandy 1000EX. Java is a favorite programming language, and Will wanted a simpler Java compiler—hence WEJAC.

Visit Will's home site:

<https://wgilreath.github.io/WillHome.html>

## **Contributing**

WEJAC is an open-source front-end to the javac compiler API. Hence if the reader wants to contribute and participate with WEJAC—it is most welcome! Test cases, bug fixes, and contributions to the WEJAC source code are encouraged.

# License

## Creative Commons Attribution 4.0 International (CC BY 4.0)

### You are free to:

**Share** — copy and redistribute the material in any medium or format

**Adapt** — remix, transform, and build upon the material for any purpose, even commercially.

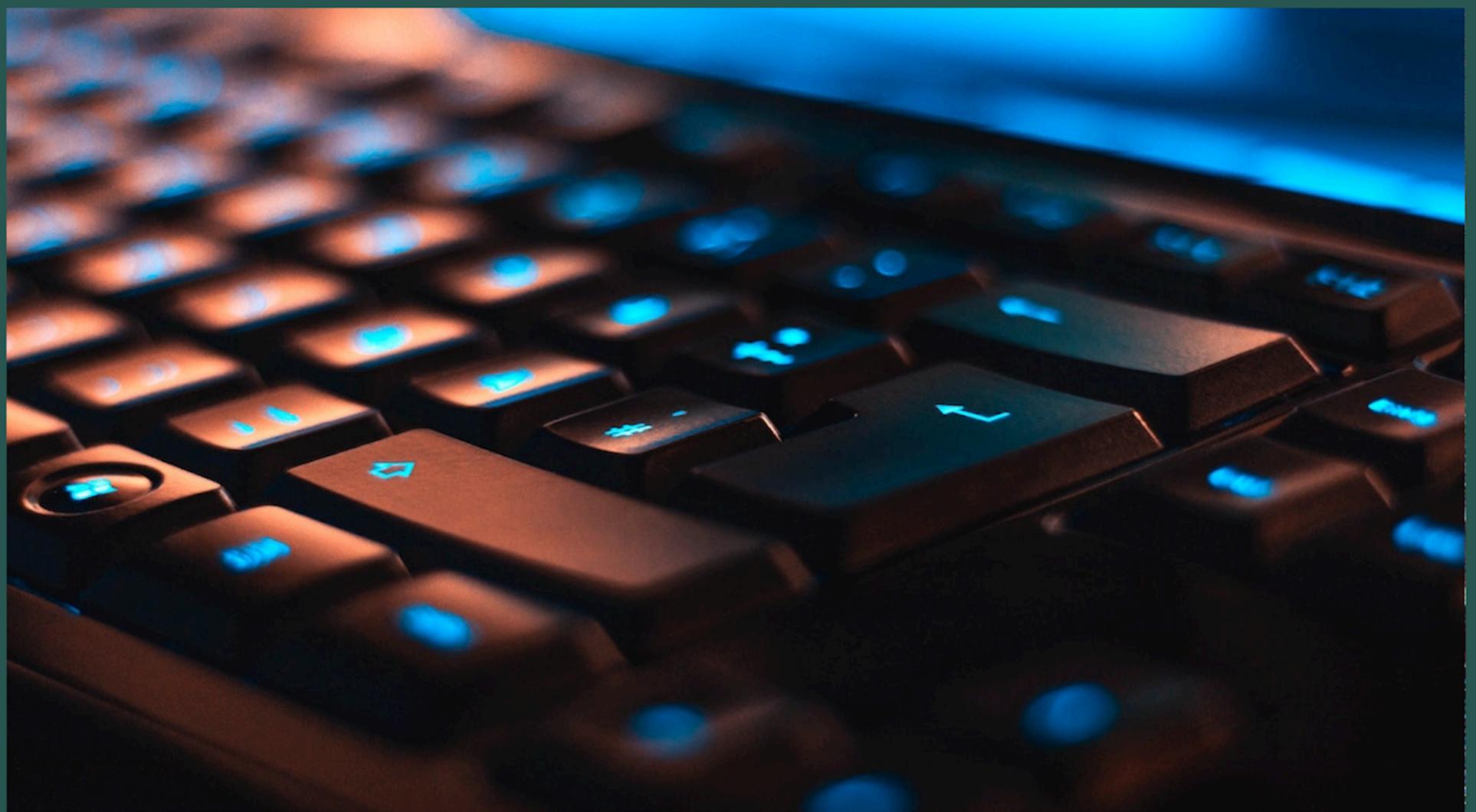
The licensor cannot revoke these freedoms as long as you follow the license terms.

### Under the following terms:

**Attribution** — You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.

**No additional restrictions** — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.

<https://creativecommons.org/licenses/by/4.0/legalcode>



## Will's Elided Java Api Compiler: a simpler Java compiler.

### WEJAC Features:

- Simplicity with fewer compiler command-line options and parameters
- Javac options and parameters can be passed to the compiler
- Java cross-platform bytecode for (macOS/Linux/Windows)
- Uses underlying Java compiler through Java Compiler API
- Source code for WEJAC is open-source under GNU GPL license v3
- Java in Java--compiler implemented using Java.
- Utilizes the Java compiler framework API (JSR 199)
- Pre-built JDK7 compatible bytecode binary JAR file
- Compiler is simple using one Java class of approximately less than 600 lines of code
- Consistent error detection and reporting across different kinds of source code errors
- Licensed under the GNU General Public License (GPL) v. 3.0
- User book is licensed under the Creative Commons share alike attribution license version 4

UserBook: <https://www.smashwords.com/books/view/948750>

Source Code Repo: <https://wgilreath.github.io/WEJAC/>

Author Home Page: <https://wgilreath.github.io/WillHome.html>

ISBN: 978-0-463-59357-8

