Installing VeriFun

VeriFun requires Java 5.0. When you enter `java -version` on your console, you should get an output like `java version "1.5.0"`. If a different version is displayed, please download and install the correct version from Sun Microsystems Inc. We strongly recommend using the **JDK 5.0** instead of the JRE 5.0, because the JDK typically includes the "server VM", which is much faster than the default "client VM".

Note: ***We strongly recommend to use the original JDK 5.0 release (displayed as "1.5.0"), as other versions of Java 5.0 have memory leaks, so that the Java VM keeps allocating memory until it eventually crashes. JDK 5.0 Updates 8 and 9 (displayed as "1.5.0_08" and "1.5.0_09") appear to have this problem. (We did not check other updates).***

Next, run the **VeriFun** setup file that you downloaded for your operating system. This will install **VeriFun** on your computer.

We recommend to install different versions of **VeriFun** into separate directories, so make sure that you select a new directory when you are asked for the setup destination.

Known problems

Windows

- Associations with `.vf` or `.omd` files do not work if the user does not have administrator privileges upon installation.

Mac OS X

- Associations with `.vf` or `.omd` files may not work. (Please let us know if they do work now with our new setup routine.) Please start **VeriFun** first and select *File/Open* to open a file.
- The *Help* command does not open the **VeriFun** User Guide. To open the User Guide, please open the corresponding PDF file that was installed during setup.

Troubleshooting

VeriFun uses the "client VM" instead of the "server VM".

Check if you have properly installed a Java version that supports the "server VM". If you get an error message when you enter `java -server -version` on the console, you might not have installed the correct Java version or your system maybe just cannot locate the correct Java version.

If you have installed a Java version featuring the "server VM", please check if the `PATH` environment variable contains the path to this Java installation. If it already contains the correct path, try to make this path the first element in the list. You can also try to set the `JAVAHOME` or `JAVA_HOME` environment variables to point to the Java installation to be used.

If this does not help, you can of course uninstall **VeriFun** and install it again. Make sure that (during setup) you select the path to the JDK that supports the "server VM". Finally, you can choose to stick to the "client VM", although it is significantly slower.

`VeriFun` crashes after allocating a huge amount of main memory.

Some versions of Java 5.0 have memory leaks, so that the Java VM allocates much more memory than specified (and needed). To solve the problem, please install a Java version that does not have this memory problem (see the Installation Section for details) and select this Java version during **`VeriFun`** setup.

Unicode

Java can be extended by more Unicode fonts. Just copy the font files into the subdirectory `/lib/fonts/fallback` of your local Java installation. Alan Wood's website gives a good overview over Unicode fonts.

Reconfiguring `VeriFun`

The steps described in this section are only necessary if you would like to change the memory configuration of* `VeriFun` *without re-installing it (and entering the desired memory configuration during setup).

`VeriFun` is executed by the Java Virtual Machine. The configuration process of the Java VM depends on your operating system and Java version. Here we describe the configuration options for the currently supported operating systems: Windows, Unix, and Mac OS X

Since the configuration of memory usage by the Java VM is a non-standard option, please consult the documentation of your local Java installation in case of problems. For instance, enter `java` or `java -X` on your console to get further information.

Windows, Unix

In the directory of the VeriFun executable you can find a configuration file with the extension `.vmoptions`. In this file, each line is interpreted as a single VM parameter. The last line must be followed by a line feed.

The following Java options can be specified:

`-Xms<size>` initial Java heap size

`-Xmx<size>` maximum Java heap size

`-Xss<size>` Java thread stack size

For `<size>` you can specify for example `5M` (for 5 megabytes).

Default settings upon installation:

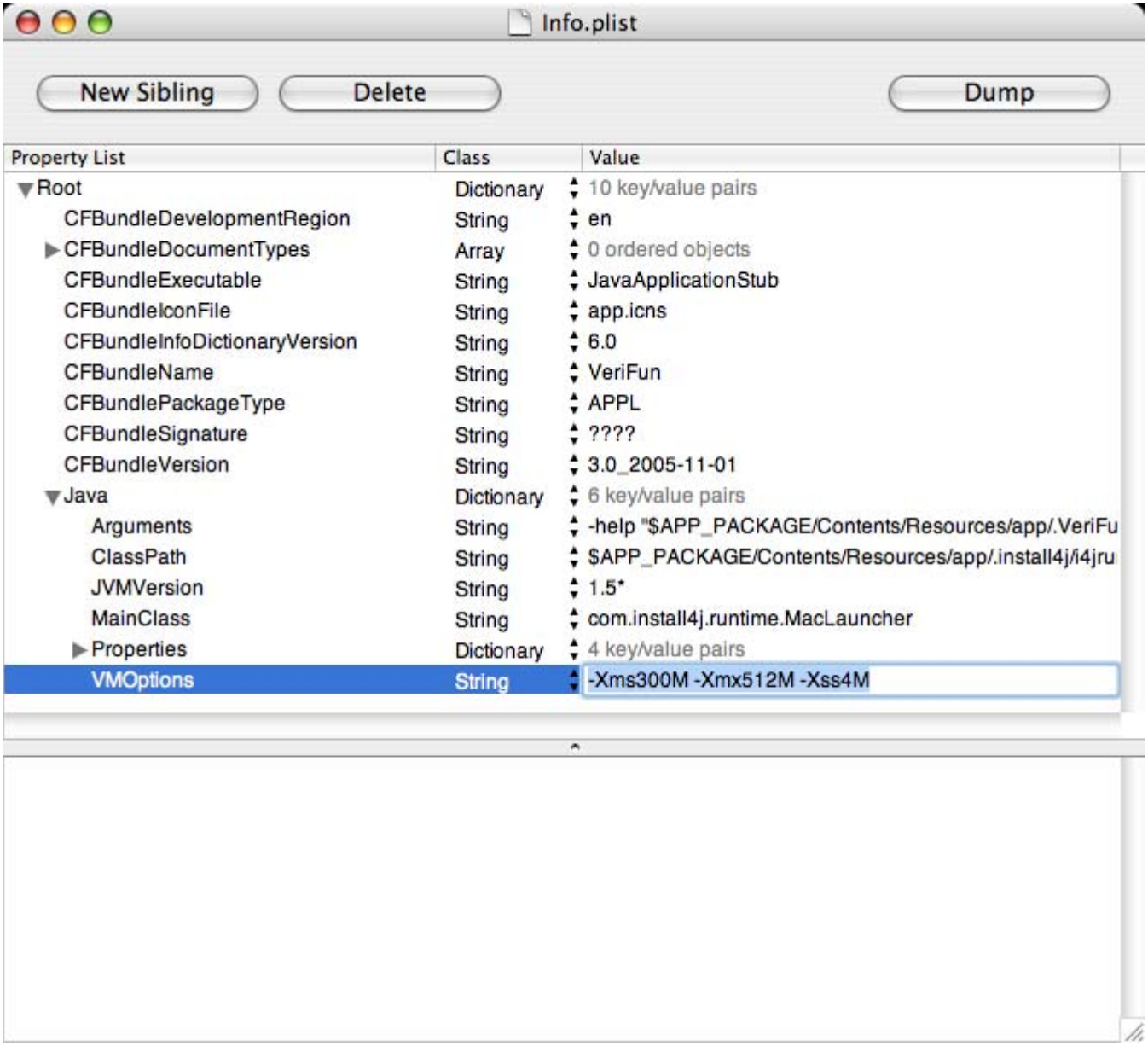
```
-Xms100M
-Xmx512M
-Xss1M
```

Mac OS X

The configuration options are similar to Windows and Unix. However, you need to edit a file called `Info.plist`. You can edit this file as follows:

(a) Using the Apple Developer Tools:

Launch the Property List Editor (in `/Developer/Applications/Utilities/`) and open the file `Info.plist`. (The open dialog of the editor considers **`VeriFun`** as a folder, so you should select *Applications* → *VeriFun* → *Contents* → *Info.plist*.) You can add the configuration options described in the Unix section to the entry *Root* → *Java* → *VMOptions*. See also the following screenshot (courtesy of Christian Hofer):



(b) Using the terminal

Without the Apple Developer Tools you need some basic knowledge of how to use the terminal window. Go to the directory that contains `Info.plist` and open the file. At the end of the file you can add or modify the configuration options:

Default settings upon installation:

```
<key>VMOptions</key>
<string>-Xms100M -Xmx512M -Xss1M</string>
```

Other Platforms

We have no experience with other platforms so far. You can ask us for help or report your experience with problems (and solutions) when installing/running **`VeriFun`** on other operating systems.