



Software Development Kit

Table of Contents

Welcome.....	3
Bem Vindo (Portuguese).....	5
Installation Instructions.....	6
For Palm OS.....	6
For Windows CE / Pocket PC.....	6
Windows 98 and beyond.....	8
Running the Samples.....	9
Using ANT.....	9
Troubleshooting.....	9
Configuring build.xml to use in your own projects.....	10
Warp and Exegen – Deploying the application.....	11
Warp Instructions.....	11
Exegen Instructions.....	12
RenameFont Instructions.....	15
Wxgn Instructions.....	15
Configuring IDEs to Debug and Run SuperWaba Programs.....	16
NetBeans or Forte.....	16
JBuilder 6 Personal and Pro.....	17
Visual Cafe (3.1).....	18
Linux	18
Eclipse 2.1	20
Microsoft Visual J++.....	20
General Settings.....	21
Optional arguments to waba.applet.Applet	21
How To Port SuperWaba.....	23
How to Contribute.....	24
Where to get support.....	25
Thanks to.....	26
GNU LESSER GENERAL PUBLIC LICENSE.....	27

Welcome

Thank you for your interest in SuperWaba. If you don't know where to start, here are some tips:

0. SuperWaba is **not Java**. Although it is 99% compatible with the Java language (the only not implemented bytecode is *synchronized*), it cannot be called a **Java VM** due to copyright restrictions. To use it, you will have to forget the JDK API and use the SuperWaba provided API (just like you would do if using J2ME). Make a good read of the API javadocs: they will show you the library differences between JDK and SuperWaba APIs.

1. SuperWaba can run in four locations: in the PDAs (currently in Palm OS from 2.0 to up and most Windows CE/PocketPC models), using a proprietary bytecode interpreter, and on the desktop, using the Sun JDK or a Java-enabled browser, or standalone, using the proprietary bytecode interpreter compiled for the Windows platform (Windows 98 or beyond).

2a. PALM OS: The VM part that needs to be installed in the Palm OS device consists of three files:

- SuperWaba.pdb: contains the classes available in SuperwabaSDK/src/waba/* and SuperWabaSDK/src/java/*.
- SuperWaba.prc: contains the Java bytecode interpreter
- SWNatives.prc: contains the implementation for the native methods (E.G.: Graphics.drawLine)
- For Palm OS 5 only: you'll need also the file 5SW.pdb (contains the fonts)

2b. WINDOWS CE: The VM part that needs to be installed in the Windows CE/PocketPC are:

- SuperWaba.pdb (same as above)
- SuperWaba.exe: same of SuperWaba.prc plus SWNatives.prc.
- You'll need also the file MSW.pdb (contains the fonts used)

3. When running the software on the desktop, you'll need a JDK (1.2.x recommended but also compatible with 1.1.x and 1.3.x). When using 1.3.x or higher compiler, you must add "-target 1.1" to the command line.

4. There are some examples in the SuperwabaSDK/samples directories. Please take a look at them. Please check the corresponding chapter in this document.

5. A User Interface tutorial, a Library (JNI-like), an Input/Output tutorial and an Optimization Tricks tutorial are available for sale at www.superwaba.org. Everyone says that they are well worth the price. Buying them helps **A LOT** for this open-source project survive.

6. Daniel Tauschke has created a free IDE for SuperWaba: MobileCreator. It simplifies a lot of the creation and building of SuperWaba programs. You can get it at <http://www.tauschke.com/>. There's also a paid library with debugging support.

7. There is also a great tool that lets you create user interfaces. You can even use it in your Palm. See /SuperwabaSDK/samples/app/GuiBuilder. It generates the code necessary to

run the application, which can be exported to the desktop to be compiled.

8. When you create a SuperWaba application (i.e. HelloPalm.java), you must create two files: HelloPalm.pdb and HelloPalm.prc/exe. The pdb will contain the HelloPalm.class file, among with any bitmap or class that HelloPalm uses. The prc/exe will contain the application version and icon, among other things. To create the pdb, use the utility superwaba/bin/Warp. To create the prc/exe, use superwaba/bin/ExeGen. The /SuperwabaSDK/build.xml file simplifies the process by compiling your program and executing both utilities.

9. If you have questions, point your newsgroup reader to news://news.falch.net/pilot.programmer.waba and ask questions there. But, PLEASE, download all messages first and do a search on them. Your question may have already been answered, and you'll save your time and the time of people at the newsgroup. (Make sure you have downloaded all messages: there are more than 20000). If you don't have a newsgroup reader, then you can use <http://groups.google.com/groups?dq=&hl=en&lr=&ie=UTF-8&oe=UTF-8&group=pilot.programmer.waba>. But note that in this case, it contains less than the half of the messages.

10. When running your application on POSE, don't forget to turn off some settings in "Settings/Debugging":

- Hardware Register Access
- Proscribed Function Call
- Screen Access

11. DON'T use the Jikes compiler with SuperWaba.

12. Be sure to read the **FAQ** at the SuperWaba site.

13. Remember: your device probably has only s 16Mhz processor. It will never behave the same as your 1000Mhz desktop computer, which usually hides most of your bad programming practices. Code optimization is always welcome. Take a look at this also: <http://patrick.net/jpt/>. And buy the Optimization Tricks Tutorial at the site.

14. Check the API javadocs at SuperWabaSDK/docs/html/index.html

15. Finally, don't forget to check <http://www.sourceforge.net/superwaba>, which hosts the sources and the VM under CVS, and contains the weekly updates.

Bem Vindo (Portuguese)

Seja bem vindo à máquina virtual SuperWaba. O SuperWaba é desenvolvido por um brasileiro. Por favor, ajude a disseminar este trabalho. A licença LGPL garante que você pode vender seus trabalhos feitos com o SuperWaba, mas quaisquer modificações na máquina virtual devem ser de domínio público e enviadas para mim com os códigos fontes, para que sejam incorporadas.

Caso você ou sua empresa precise, nós lecionamos cursos de SuperWaba, de forma que se possa tirar o máximo da plataforma. Além disso, prestamos consultoria técnica na área de computação móvel e temos ótimos planos customizados de suporte para a plataforma, inclusive com auxílio telefônico. Existem contratos disponíveis no site, bastando se cadastrar como brasileiro e fazer o login.

Caso tenha dúvidas técnicas, entre no grupo de discussão oficial do SuperWaba para brasileiros: <http://br.groups.yahoo.com/group/superwaba/>

Leia também o artigo escrito para o site Clube Palm: <http://www.clubepalm.com.br>

Installation Instructions

For Palm OS

Please install the following files in your device:

- For Palm OS < 5 – Files located in folder **lib/palm/PalmOS2_3_4**
 - SuperWaba.prc (VM)
 - SWNatives.prc (VM)

- For Palm OS >= 5 and Palm OS 4.1 16bpp_or_higher devices that can't emulate 8bpp (including Tungsten W and Kyocera 7135) – Files located in folder **lib/palm/PalmOS5_or_16bpp_only**
 - SuperWaba.prc (VM)
 - SWNatives.prc (VM)
 - 5SW.pdb (Fonts)

- For all Palm OS versions – File located in folder **lib/xplat**
 - SuperWaba.pdb (API Classes)

2. When installing a SuperWaba program, be sure to install both prc and pdb files generated.

3. If you're using files from the ext package, install the corresponding library file (located in folders **/lib/xplat** and **/lib/palm**). Some samples also need an extension library to run. Check the corresponding chapter in this document.

4. When creating your own programs, be sure to use the same Creator ID in both EXEGEN and WARP (or just use the default one). Otherwise, the program won't run.

For Windows CE / Pocket PC

0. Run the CEinstall_RunMe.bat file located in folder **lib/ce** OR follow the instructions below:

1. Please install the files
 - SuperWaba.exe (VM)

- SuperWaba.pdb (API Classes)
- MSW.pdb (Font file)

...in your WINDOWS CE / POCKET PC device under the **/SuperWaba** directory (from the root).

2. You must choose the right EXE file located under the directories, according to the processor and which platform you're using (those EXEs are located under the directory **/lib/ce**).

3. File SuperWaba.pdb is located in folder **lib/xplat**. File MSW.pdb is located in folder **lib**.

4. When installing a SuperWaba program, be sure to install both EXE and PDB files generated with Exegen and Warp. The EXE file must be installed in the **Start Menu**, ideally under a folder called **SuperWaba**, along with other SuperWaba programs.

5. The PDB file must be installed in a directory specific for your programs. You **cannot** install it in the **/SuperWaba** directory. The directory is specified when running the EXEGEN program and is, by default, the same name of your Main class.

6. If you're using classes from the ext package, install the corresponding library file (located in folders **lib/xplat** and **lib/palm**). Some samples also need an extension library to run.

7a. SuperWaba global libraries must be installed in the **/SuperWaba** directory.

7b. SuperWaba local libraries must be installed in the application's directory.

7. You can also install optional fonts. Remember, all fonts must be placed under the **/SuperWaba** directory.

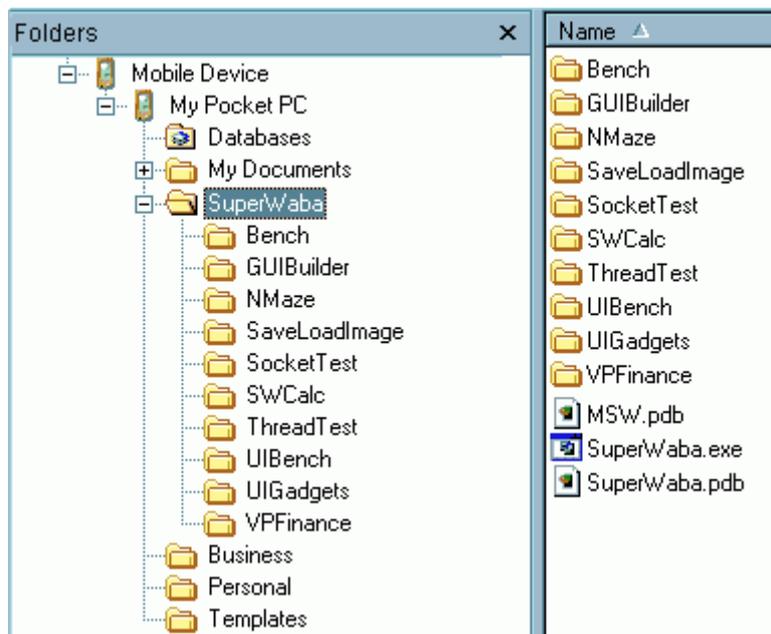
8. When creating your own programs, be sure to use the same Creator ID in both EXEGEN and WARP (or just use the default one).

9. If you don't follow these rules, your program will not run.

10. To run the welcome application on Windows CE, click in the SuperWaba.exe file using the File Explorer or create a SuperWaba.lnk file with the following line, installing it in the **Start Menu**:

```
24#\SuperWaba\SuperWaba.exe
```

11. The picture below shows an example of directory organization:



Windows 98 and beyond

1. Create a new folder, named c:\SuperWaba
2. Copy the files **lib/win32/SuperWaba.exe**, **lib/xplat/SuperWaba.pdb** and any other library needed (**PDB** and **DLL** files, not JAR or CLASS files).
3. Install **one** of the two font files in c:\SuperWaba: 5SW.pdb or MSW.pdb. Its possible to emulate both Windows CE and High-resolution Palm OS PDAs (“/w 240 /h 320” for the first one and “/w 320 /h 320” for the second one – default). **Do not install both font files.**
4. Create a directory for your program under c:\SuperWaba, just like in the Windows CE/ Pocket PC targets (as explained in the last topic), place its PDB files there.
5. To run the VM, double-click in the SuperWaba.exe file.
6. To run your program, create a .bat file, a .cmd or a shortcut. The bat/cmd files must be placed in the c:\SuperWaba folder. Below is an example for the .bat or .cmd files:
 - c:\SuperWaba\SuperWaba.exe UIGadgets UIGadgets DoHD
 - Runs the program UIGadgets, located under c:\SuperWaba\UIGadgets folder
 - c:\SuperWabaSDK\lib\win32\SuperWaba.exe UIBench
c:\SuperWabaSDK\samples\app\UIBench\build OhdB
 - Runs the program UIBench, telling that the UIBench.pdb is in the given path.
 - It also specifies that the SuperWaba.exe file is not in the default place
 - c:\SuperWabaSDK\lib\win32\SuperWaba versapalm/finance/VPFinance VPFinance AnGD
 - Runs the versapalm.finance.VPFinance program, located under c:\SuperWaba\VPFinance folder.
7. A shortcut can be created with the same parameters like above in the *object*, and the “*start in*” option must contain c:\SuperWaba (or other folder where the SuperWaba.pdb and other files are placed).

Running the Samples

Using ANT

There are some samples in the folder **samples**.

To run the sample, you can use the build.xml ANT makefile. You may download ANT from here: <http://ant.apache.org/>. If Ant is not running after you unpack it, set the **java_home** environment variable pointing to where you installed JDK (e.g.: set java_home=c:\jdk1.2.2).

Each sample has a build.xml file, with application specific settings, that calls the base SuperWabaSDK/build.xml.

Open a shell (if running on Windows, go to the command prompt), change to the SuperWabaSDK folder, and type "ANT help" to see the instructions.

To rebuild all samples, use "ANT BUILD-SAMPLES" under the SuperWabaSDK folder.

Troubleshooting

If, after you install the sample in the device, an error message appears saying "Can't find class", you'll need to install also a SuperWaba Extension Library. Next is a list of the samples and the required libraries:

- app/Guibuilder: XPlatUiUtil.pdb, PalmloBuiltIn.pdb
- app/Watch: PalmloBuiltIn.pdb
- game/Ping: XPlatGame.pdb, XPlatUtilProps.pdb
- io/Socket: XPlatUiUtil.pdb
- io/HtmlViewer: XPlatHtml.pdb
- io/PalmDoc: PalmloDoc.pdb, XPlatUiUtil.pdb
- io/PDBConduit: XPlatUiUtil.pdb
- io/Scanner: PalmloScanner.pdb
- ui/Grid: XPlatUiUtil.pdb
- util/Xml: XPlatUtilXML.pdb
- util/Zip: XPlatUtilZip.pdb, XPlatUiUtil.pdb

Configuring build.xml to use in your own projects

To use ANT for your own programs, just copy one of the build.xml files (for example SuperWabaSDK/samples/app/SWCalc/build.xml) to your directory. The settings that must be changed to match your application's are:

Target **compile-init**

- `<property name="sw.root" value="{basedir}/../../.."/>` : change the value to point to SuperWabaSDK folder

Target **build-init**

- `<property name="app.name" value="SWCalc"/>` : *change to your application's name.*
- `<property name="app.creatorid" value=""/>` : *set to your application's creator id, if not the default one.*
- `<property name="version" value="2.2"/>` : *set your application's version.*
- `<property name="warp.includes" value=""/>` : *specify any include path for warp.*
- `<property name="warp.excludes" value="superwaba"/>` : *specify any exclude path for warp.*
- `<property name="main.class" value="SWCalc"/>` : *specify the full package name for your application.*

Target **run-init**

- `<property name="applet.scale" value="1"/>` : *specify the scale when running as applet (1,2,3,4).*
- `<property name="applet.bpp" value="8"/>` : *specify the bpp for emulation (2, 4 or 8)*
- `<property name="applet.width" value="320"/>` : *specify the desired width (160,240,320,480,640)*
- `<property name="applet.height" value="320"/>` : *specify the desired height (160,240,320)*
- `<property name="applet.uiStyle" value="WinCE"/>` : *specify the user interface style (WinCE or PalmOS)*
- `<property name="applet.useSonyFonts" value=""/>` : *specify if its to emulate Sony devices with high resolution but Palm OS < 5 (" or "useSonyFonts")*

Warp and Exegen – Deploying the application

The files described here are located in the **utils** directory.

These files were taken from the WExtras class, package wababin, and enhanced for SuperWaba 2 and above.

Copyright 1999-2001 Rob Nielsen

Copyright 2002-2003 Guilherme Campos Hazan

To learn how to compile your applications using them, please read the faq at the site.

There are 4 callable files in this directory:

- . warp - packages your class files into a pdb
- . exegen - creates an "executable" file that calls the vm inside the device
- . RenameFont - renames a font pdb file so it fits in the SuperWaba Font-Name Pattern.
- . Wxgn - calls warp and exegen with only one command

Warp Instructions

Usage: java Warp command [options] warpfile [files]

Commands (case insensitive):

- c Create new warp file
- l List contents of a warp file

Options (case insensitive):

- /? Displays usage text
- /c Override and assign PDB database creator (e.g. /c CrTr)
- /r If a directory is specified in the files, recurse any subdirs
- /q Quiet mode (no output except for errors)
- /\$ Set the Palm OS copy protection attribute
- /x Specify a list of package/class names whose classes won't be added.
You can specify a list without spaces as 'class1,class2/,package1,...'
 - . The files whose full package name starts with the given string
 - . The files whose class name starts with the given string
 - . If string ends with \, only the whole class name is used for compare

E.g.:

`/x extra,jgui.sio.server,NativeMethods4JDK,Graphics\`
-> packages `extra.*` and `jgui.sio.server` are excluded,
-> all class names starting with `NativeMethods4JDK` are excluded,
-> `Graphics` class is excluded (but `GraphicsNat` is included)
`/i` Specify a list of package/class names whose classes will be the only ones that will be included. The opposite of `/x` option.
`/li` Specify that this file will be a SuperWaba Local Extension Library.
`/lg` Specify that this file will be a SuperWaba Global Extension Library.

This program creates a `.pdb` file, used on all SuperWaba platforms. The creator will be generated automatically from the name of the `pdb` file if not overridden with the `/c` option.

The input file can be a Jar (in this case, all `.class` and `.bmp` files in the jar will be added).

Warp will automatically check any class files for dependencies and add these files so you will only need to specify the main class file and everything else will be added automatically, even directly referenced `.bmp` files. (ie. `Image im=new Image("rob.bmp)`
If no input files are specified, it will look for a `.class` file with the same name of the `pdb` file you are creating.

Examples:

```
java Warp c HelloApp
java Warp c /c Crea HelloApp helloapp.jar
java Warp c helloApp *.class util\*.class
java Warp c /r helloApp extra\
java Warp c /lg wextrasLib *.class extra\
java Warp l helloApp.pdb
```

Exegen Instructions

Usage: `java Exegen [options] exefile [main-window-class] [pdbfile]`

Options (case insensitive):

`/?` Displays usage text
`/$` Turns on the PalmOS copy protection attribute
`/C` Override and assign PDB creator (e.g. `/c CrTr`)
`/H` Assign height of application's main window
`/W` Assign width of application's main window
`/P` Subdirectory name for the application under Windows CE. Note that in SuperWaba all applications stay under SuperWaba at '\SuperWaba' Spaces, `/`, `\` and `"` in the directory name are not allowed.
`/V` Assign the version for the application
`/Z` Creates CAB installation files to deploy your application
`/E` generates `.exe` files for Windows CE and Pocket PC.
`/L` generates `.lnk` file for Windows CE and Pocket PC.
`/I` Specifies the icon PREFIX.

This program generates a Palm OS `.prc` and Windows CE application `.exe` files
These executables are used to launch (start up) a SuperWaba program.

If you don't specify the `pdb` file, its name will be the same of the `exe` file.
If you don't specify the `[main-window-class]`, its name will be the same of the `exe` file too

File extensions are generated automatically. For example, if you specify `myapp` as the `exe` file, a `myapp.exe` and `myapp.prc` will be created.

The `/W` and `/H` parameters define the default width and height of the application's window. The value of 0 for either will cause the main window to appear at a default size which is different on each platform.

If no icon is defined, a default one is used. Icons are scanned first in the current directory, next in the classpath. You may provide the following files (WIDTHxHEIGHTxBPP): `icon15x9x1.bmp`, `icon15x9x8.bmp`, `icon16x16x4.bmp`, `icon22x22x1.bmp`, `icon22x22x8.bmp`, `icon32x32x1.bmp`, `icon32x32x4.bmp`, `icon32x32x8.bmp`, `icon48x48x8.bmp`, `icon30x18x1.bmp`, `icon30x18x8.bmp`, `icon44x44x1.bmp`, `icon44x44x8.bmp`. If you specify a prefix (E.g.: AAAA), the searched files will be `AAAAicon22x22x8.bmp`, etc.

The 15x9, 22x22, 30x18 and 44x44 are for Palm OS (last 2 are for double density displays, E.G. Palm OS 5 and Sony Clie), and the others for Windows CE.

Note that for Palm OS you must create bitmaps with the web-safe-palette.

A gif with this palette is located under SuperWabaSDK/samples.

For Windows CE, the palette CANNOT BE THE WEBSAFE one. This happens because in the Web safe palette, index 0 is white, and WinCE devices expect that index 0 will be black. A warning will be issued if you use an incorrect palette when generating the exe files for WinCE. To correct this problem, you may use any image editor to increase the image to 16bpp or more and then decrease it again to 256 colors and save the image. Most editors don't use the web safe palette, which will correct the problem.

For best compatibility, copy the icon*.bmp files in SuperWabaSDK/utls to your application's directory and edit them.

A PalmOS PRC creator and PRC name will be assigned based on the class name and exefile respectively. The exefile must be 30 characters or less.

The /Z parameter is used to create eight cab files so your application can easily be installed on all compatible Windows CE platforms. You may pass, optionally, a .swz file with some global and local libraries used by your app. For example, if you need to install the TinyLarge.pdb font and also a custom database, E.g. mydata.pdb, you must copy all files to the current directory, then create a myapp.swz file with the following contents:

```
[G]TinyLarge.pdb
```

```
[L]mydata.pdb
```

The [G] denotes a global library (fonts are always global libraries), and the [L] denotes a local library.

The user will then have two options to start the installation process:

1. Run the created xxx_install.bat file, or
2. If the cab files are available on the internet, the user may launch the browser from inside the device, connect to the site and choose the appropriate cab file. The browser will then download and install the file.

The /Z parameter works only on the Windows platform.

Examples:

```
java Exegen /I myicons.ico /P Scribble scribble
```

```
java Exegen /W 160 /H 160 Calc CalcWindow calc
```

```
java Exegen /Z SWCalc.swz SWCalc
```

👁 After you install the new icon on the device or in the emulator, a soft reset is needed to show it up.

RenameFont Instructions

Usage: RenameFont <fontpath> <oldfontname> <newfontname>

Please dont append the .pdb extension to the font names.

Note that the original files are preserved.

Wxgn Instructions

Usage: java Wxgn Warp_commands z Exegen_commands

Note that z is the separator

Configuring IDEs to Debug and Run SuperWaba Programs

Below you see some ways of how to set up some famous IDEs to run and debug SuperWaba programs as applets or applications in the desktop. After you debug it, you can generate the necessary files with Warp/Exegen and deploy to the device.

NetBeans or Forte

Thanks to Jean Rissoto, Ed. J Szalajeski and Vince Nguyen

In NetBeans (or Forte)

- Click the Tools menu, select Options and go to the Debugging and Execution folder:
- Expand the folder
- Select the "Execution Types" icon (looks like 2 arrows) and expand the tree.
- Right Click your mouse on the root of the Execution Types, and select New.. External Execution Name the Type ExternalSuperWaba
- Right Click the ExternalSuperWaba Item just created, from the context menu select Properties, and change the External Process to
`-cp {filesystems} waba.applet.Applet /datapath . /x 10 /y 10 /scale 2 /bpp 8 {classname}`
- Select the Debugger Types tree.
- Expand the tree
- Right Click your mouse and select new Default Debugger. Name it DefaultSuperWaba
- Right Click the DefaultSuperWaba just created, from the context menu select Properties
- Change the External Process Argument to
`{classic}{debuggerOptions} -Djava.compiler=NONE {q}{bootclasspathSwitch}
{bootclasspath}{q} -classpath {q}{filesystems}{q} waba.applet.Applet /datapath . /x 10 /y 10 /scale 2 /bpp 8 {main}`
- (please note: you are only adding waba.applet.Applet to the default)
- Close the options menu,
- In the filesystem Explorer
- unmount all jar file netbeans add by default
- mount Sun's source jar file for the rt.jar (src.jar) otherwise you will not be able to debug
- mount the directory where SuperWaba classes are. This allows stepping through the SW source code
 - X:\SuperwabaSDK\src
- An alternative is to mount only the SuperWaba.jar file (/SuperWabaSDK/lib/superwaba.jar), and not directly step into the SW libraries. Or create a jar with both .java and .class files.
- mount your own jar files or java files

Project settings:

- Select your project's mount directory in the filesystem explorer, (the directory with your source files)
- Right Click your main class file, select properties from the context menu
- Select the Execution Tab,
- Change the Compiler to Internal Compilation
- Change the Debugger to DefaultSuperWaba (your item created above should be in the drop down combo box)

- Change the executor to ExternalSuperWaba
- The arguments can be adjusted from this property page too.
- (Recommended arguments)
- Arguments: /dataPath j:\apps\my\donnees /x 10 /y 160 /bpp 8 /scale 2
j:\apps\my is the path to the datas of my app...
/x 10 /y 160 /bpp 8 /scale see explanations below at end of text.

JBuilder 6 Personal and Pro

- 1) Create a new project
 - 1.a) Select the menu "File"->"New Project"
 - 1.b) Set the Name of the project: for this example, "SWTest" will be the name of the project:
- 2) Create/Add a "Lib":
 - 2.a) Right click on SWTest.jpx
 - 2.b) Select Paths
 - 2.c) Select Required Libraries
 - 2.d) Select Add -> New (1st time only)
 - 2.e) Set Name to SWLib (for instance), then Select Add
 - 2.f) Select superwaba.jar (SuperwabaSDK/lib/superwaba.jar)
 - 2.g) Select your new Lib (SWLib) and select OK
 - 2.h) you're back to the "Path" tab of the project properties window.
- 3) Select your Lib again and click on Edit. You will get the Configure Lib window.
- 4) Select Source -> Add and select your SuperwabaSDK/src folder
- 5) You can also set the "documentation" path to SuperWabaSDK/docs/html.
- 6) Select OK to close the Configure Lib window and the project properties window

Now, you need to modify the following only for SuperWaba and restore it after for a regular project.

- 7) Select, from the menu bar, Tools->Configure JDKs
- 8) Add and move to the top the following paths (in this order, from the top to the "bottom"):
 - 8.a) SuperwabaSDK/src
 - 8.b) SuperwabaSDK\lib\superwaba.jar
- 9) You can add the last 2 paths to the source and doc paths too by selecting the available tabs
- 10) Select OK to close this Configure JDKs window.

You're done and ready to do everything you need now. The trick is that you have to execute/debug the waba.applet.Applet class with the Name of your main window subclass as the application parameters. Be sure to run/debug waba.applet.Applet as an application and not as an applet:

Put a breakpoint on the first line of your onStart method and launch Debug. It will stop where you did set your breakpoint. If you don't do that, you will debug waba.applet.Applet, which is probably not what you want to do.

Visual Cafe (3.1)

1. menu File/New project. Choose "empty project".
2. menu Project/Options.
 - Project tab:
 - . Release type: "debug".
 - . Project type: "Application"
 - . Main class: "waba.applet.Applet"
 - . program arguments: <your_app_name>
 - Directories Tab:
 - . Input Class Files: add in this order
 - . <visualCafe_directory>\java\Lib\classes.zip
 - . SuperwabaSDK/lib/superwaba.jar
 - . <your_project_directory>
 - . uncheck "Append classpath"
 - . uncheck "Auto-generate classpath"
 - . Source files: add
 - . SuperWabaSDK/src
3. press ok.
4. drag your .java project files to the project pane.
5. menu Project/Build Application
6. set your breakpoints
7. menu Project/Run in Debugger

Note: when you change the Project release type for "final", you must set again the paths.

Linux

Thanks to Christopher C. Stump

Establishing a SuperWaba development environment under Linux is a rather easy task. All the tools necessary for SuperWaba development are available in the SuperWaba SDK which can be downloaded from www.superwaba.org. Once the package is obtained, it needs to be unzipped.

```
#Unzip the package  
[command prompt]$ unzip SuperWabaSDK.zip
```

Once the SDK is extracted, it needs to be placed in a useful location. We will assume the environment is being established for a single user, in which case anywhere in the user's home directory is acceptable.

```
#Move SDK to home locale  
[command prompt]$ mv superWabaSDK ~/any location
```

Once the SDK is in its desired location one must set the Java classpath environment variable to include the new library. In a single user environment, a personal start-up script can be used for this purpose, such as `~/.bash_profile` which is provided by any Linux system that gives its users a Bash shell by default. You can temporarily define the classpath variable in a shell, but it is much more useful to define it permanently in the

above file. In any case, the following must be defined somewhere on the system:

```
CLASSPATH="{full path to the SDK}/lib/superwaba.jar:{full path to SDK}/superwaba/utils:."
```

The superwaba.jar holds the SuperWaba class files. The utils directory holds tools that will be used during execution of Warp and Exegen. If a classpath variable is already defined on the system, then the above path can be appended to the variable with the same syntax. Also, note the last ':.', this is necessary to include the current working directory in the classpath. Without this addition, compilation will fail. In order for the new classpath variable to take effect, the shell or X window environment being worked in must be exited and the system logged back into.

With the SuperWaba SDK in place and the classpath properly defined, one can begin developing SuperWaba applications. We will assume the reader has already written a SuperWaba program called MySuperWabaApp.java . In order to test/run this program either on a Palm OS device or POSE, we must transform the .java source code into a Palm OS executable .prc file. This can be accomplished in a few easy steps:

```
#Change to the directory with the SuperWaba source  
[command prompt]$ cd {path to SuperWaba source}
```

```
#Compile source  
[command prompt]$ javac MySuperWabaApp.java
```

```
#Run Warp on the newly created .class files  
[command prompt]$ java Warp c MySuperWabaApp MySuperWabaApp.class
```

```
#Run Exegen on the .class files  
[command prompt]$ java Exegen MySuperWabaApp MySuperWabaApp  
MySuperWabaApp
```

Warp is a java program found in {full path to SDK}/lib. Warp's purpose is to create the .pdb files used by Palm OS applications and the .wrp files used by Windows CE devices. We are only concerned with the .pdb file. The first argument to Warp, c, tells the program to create a .pdb file. The second argument is the name of the to-be-created .pdb file, and the third argument is all of the .class program files.

Exegen is a java program that can also be found in same directory as Warp. Exegen's purpose is to create the .prc file, which is the SuperWaba program in Palm OS executable form. The first argument to Exegen is the name of the to-be-created .prc file. The second argument is the name of the main source code file. The third argument is the name of the .wrp file.

If the above steps were successful, then a ready-to-use/test Palm OS application should be in the current directory. There should be files called MySuperWabaApp.prc and MySuperWabaApp.pdb . Both of these files, along with the SuperWaba virtual machine files, must be installed on a Palm unit or POSE in order to run the application. The SuperWaba virtual machine files can be found in {full path to the SDK}/ . Of course, the above command line arguments can be tedious to type over and over again while developing. Therefore, it is much nicer to place all of the commands in a simple shell script, like this Bash script, so everything can be easily run in sequence with a single command:

```
#Invoke shell script to handle SuperWaba app creation
[command prompt]$ ./superwaba_maker.sh
```

Eclipse 2.1

Created by Guilherme Campos Hazan. For example, we will setup to debug the UIGadgets example.

To create the project:

1. Open menu File/New/Project/Java/Java Project. Click next.
2. Choose a name for your project. (E.g.: UIGadgets)
3. Uncheck the "Use default" option and open the directory where the application is. (E.g.: SuperWabaSDK/samples/ui/UIGadgets)
4. Click next. Press yes when asking to create project now.
5. Select the Libraries panel. Click "Select External Jars", and add the SuperWabaSDK/lib/SuperWaba.jar file.
6. Click finish.
7. Right-click the project, choose Properties.
8. Select the "Java Build Path" page, then select the Libraries tab.
9. Click in the + near the SuperWaba.jar file to open the options.
10. Double-click the "javadoc location". Select SuperWabaSDK/docs/html directory.
11. Double-click the "source attachment". Select the same SuperWaba.jar file.
12. Click ok to finish.

To debug/run the project:

1. Click menu Debug/Debug... or Run/Run...
2. Click New
3. Enter the name of the configuration. (E.g.: UIGadgets)
4. In the "Main" tab:
 - Project: click browse and select your project name (E.g.: UIGadgets)
 - Main class: type "waba.applet.Applet"
5. In the "Arguments" tab:
 - Program arguments: type the class name that extends waba.ui.MainWindow (E.g.: "UIGadgets")
 - You may also enter, before the class name, the other possible parameters (see last topic of this chapter)
6. Click Run / Debug.

Thats it! You're now able to debug your projects in Eclipse!

Microsoft Visual J++

1. Put SW classes in the first position in the classpath. E.g. if your classpath is `SET CLASSPATH = C:\jdk1.2.2` now becomes `SET CLASSPATH=C:\SuperWabaSDK\lib\superwaba.jar;C:\jdk1.2.2`
2. Open in Visual J++: File < New Project < Console Application

3. OK now you can start to write code.
4. To run, test and debug, go to Project > "app" Properties... Launch TAB, check Radio Button "Custom"
5. Set program: JVIEW.EXE and arguments: waba.applet.Applet "ARGS" "yourclass" (where ARGS are the arguments to pass to SuperWaba Applet and "yourclass" is the name of you main class. E.g.: waba.applet.Applet /scale 1 firstScreen)
6. Now Tab Output Format: uncheck Enable Packaging

PS (added by Freddie Oliveira): CLASSPATH environment variable is lost when J++ IDE calls JView. This problem is reported by Microsoft in article "PRB: Environment Variables Are Lost When Running from VJ6 IDE" (Q215413).

To solve the problem, follow all steps in SW faq and add the CLASSPATH env variable contents in registry key: HKLM\SOFTWARE\Microsoft\Java VM\Classpath. Eg.:
 HKLM\SOFTWARE\Microsoft\Java VM\Classpath = %systemroot%\java\classes;.;c:\SuperWabaSDK\lib\superwaba.jar;c:\j2sdk1.4.1_01\lib\tools.jar.

General Settings

To be able to do the setup in any IDE, you must do as described:

1. Add to the classpath your project directory
2. Add to the classpath "SuperwabaSDK/lib/superwaba.jar" file. Maybe you'll need to recompile the classes or remake the jar file if you're not using JDK 1.2.2, that is the jdk used to deploy SuperWaba.
3. Set the main class as "waba.applet.Applet"
4. Set the program arguments to your class name with full package specification. For example, to run the Welcome application as an applet, set the command line to "waba.ui.Welcome"

Optional arguments to waba.applet.Applet

(when running it as application),
 in any order and case insensitive:

- . /x <x>
- . /y <y>
- . /w <width>
- . /h <height>
- . /uiStyle <WinCE_or_PalmOS>
- . /bpp <bitsPerPixel_2_4_or_8>
- . /scale <scale_1_to_4>
- . /dataPath <path>
- . /useSonyFonts
- . /t title of the application window
- . /cmdLine <the rest of arguments are passed to the app as the commandline>
- . class name must always be the last argument

Note: The dataPath is used to set where the catalog and images are stored

Note2: In version 2.0beta4, appPath was changed to dataPath to make more sense, but appPath is still used for backwards compatibility

If you want to emulate a Sony 320x320 device, use the following parameters:

```
rundemo /w 320 /h 320 /scale 1 /useSonyFonts /bpp 8
```

To emulate a Palm OS 5 device (like Tungsten T), use:

```
rundemo /w 320 /h 320 /scale 1 /bpp 8
```

To emulate a Windows CE or Pocket PC device with 240, 480 or 640 horizontal resolution, use:

```
rundemo /w 240 /h 320 /scale 1 /bpp 8
```

```
rundemo /w 480 /h 240 /scale 1 /bpp 8
```

```
rundemo /w 640 /h 200 /scale 1 /bpp 8
```

How To Port SuperWaba

The src and VM files are in the <http://sourceforge.net/projects/superwaba> CVS repository and also in the SDK under the SuperWabaSDK/src and SuperWabaSDK/vm. Sourceforge also contains an almost nightly update of the sources.

Go to **vm/xplat**. There are some files with some methods that must be filled in order to port the VM (note: the files presented there are a little out-of-date). Take a look also at the **vm/ce** files, because their routines are easy to understand and are updated.

Be sure to check also SuperWaba.h file, where you can define, like the platform define which will select which files and functions will be compiled.

Search, in every file, for "fill in" and you'll find the appropriate places to fill in with the device specific code.

The class loader, font management, and waba.io.Catalog will be automatically implemented when you port the Catalog.c device specific functions (only eight functions, and that's it!). They all rely on those methods.

If you have any problems, feel free to contact us at support@superwaba.com.br

Good luck, and may the force be with you!

How to Contribute

1. Write great software with SuperWaba, or an article, or a white-paper, or a book, or more libraries, or more documentation, or a how-to...
2. Publish your site to PalmGear (www.palmgear.com). This helps spread SuperWaba.
3. Buy tutorials (User Interfaces, IO, Library and Optimization Tricks) at the site.
4. Buy the external libraries. Currently the PDBDriver, CatalogSearch and the Printing Library are available.
5. Buy support. Answers available in 4 hours or less.
6. Port the VM to other operating systems. Just fill in some methods and thats it!
7. Publish useful source code and also links to your programs at the **programs** area of the SuperWaba site.
8. Place ads of your SuperWaba products at the SuperWaba site. Very low rates!

Where to get support

To get support, you can visit

www.superwaba.org
www.wabaworkbench.com
www.wabasoft.com/howto.shtml

Also, if you have questions regarding the VM, point your newsgroup reader to

`news://news.falch.net/pilot.programmer.waba`

But before posting a question, download all messages from the newsgroup (+10000) and do a search in the messages already posted; some questions come up over and over again once a week.

Be sure to read the SuperWaba faq at SuperWaba.org site.

Buy the tutorials available at the site.

Buy also custom support: answer in 4 hours or less.

Thanks to...

All SuperWaba community who are always helping to improve this software.

Special thanks to:

- Rick Wild: who created Waba and started everything
- Tom Cuthill: who provided the Scanner support
- Eric Frick (AirNet Systems) who donated a Symbol SPT 1500
- Dave Slaughter: for the 1.21 WinCE/desktop versions
- Diego Montalvo, the webdesigner of the SuperWaba.org site
- Ed Crandell, creator of the helpful WabaWorkbench site
- Rob Nielsen, creator of WExtras and the java version of exegen/warp
- Mike Aeneas, who is constantly helping me in various issues
- Mike Brereton, author of EWE, who kindly helped me every time I asked for help in the Windows platform
- David Britto, who presented me to the Java and the Palm worlds
- A sincere thanks to Quality Software (www.quality.com.br) who permitted me to work full-time on SuperWaba during 3 months when I was an employee
- Peter Dickerson, who is always helping improve SuperWaba's reliability.
- David Haysmith (from The WhiteClarkeGroup), a valuable SuperWaba partner
- Sir Andy Faeryblood, who provided important improvements for the VM (popup blocking modal windows, and the db2e library)
- Arnaud Farine, for the great support in France.
- Jean Rissoto, for the MultiEdit class and also many other VM improvements.
- Frank Diebolt, for the great work in the game package and the complete setup of SourceForge.
- Daniel Tauschke, author of the first IDE dedicated to easy SuperWaba development/deployment.
- Osvaldo Garcia and Renan Bundt (from Copel S.A.) for the donation of a Kyocera QCP 6035.

GNU LESSER GENERAL PUBLIC LICENSE

Version 2.1, February 1999

GNU LESSER GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does. 1. You may

copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) The modified work must itself be a software library.

b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably

considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work

that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified

materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who

have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software

distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

Subsection 6 of the above mentioned terms and conditions specifies that, "You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License."

For the purposes of the SuperWaba software we **request** that software using or linking to the SuperWaba virtual machine or its libraries display the following notice: "**Created with SuperWaba www.superwaba.org**"

This request of notice is consistent with the terms and conditions stated in the GNU Lesser General Public License version 2.1.