

# Einstein Platform User's Manual

For Open Einstein 2007.7

July 9, 2007

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Requirements</b>	<b>4</b>
<b>3</b>	<b>Extraction of the ROM from your Newton</b>	<b>5</b>
3.1	Hammer/Newtsbug (serial line) . . . . .	5
3.2	ROM Dumper (TCP/IP) . . . . .	5
<b>4</b>	<b>Einstein on MacOS X</b>	<b>8</b>
4.1	Using the CLI flavor on MacOS X . . . . .	8
4.2	Options . . . . .	8
<b>5</b>	<b>Einstein on arm-linux PDAs</b>	<b>10</b>
5.1	Quickstart on the Zaurus . . . . .	10
5.2	More details for other PDAs . . . . .	10
5.3	Options . . . . .	11
<b>6</b>	<b>Developer notes</b>	<b>12</b>
6.1	Compiling and porting Einstein . . . . .	12
6.2	CLI commands . . . . .	12
6.3	Monitor mode . . . . .	12
6.4	Logging . . . . .	13
<b>7</b>	<b>Relativity Developer Guide</b>	<b>14</b>
7.1	Introduction . . . . .	14
7.2	Sample code . . . . .	15
<b>8</b>	<b>Known problems</b>	<b>17</b>
<b>9</b>	<b>Changes History</b>	<b>18</b>
<b>10</b>	<b>License</b>	<b>19</b>

## **1 Introduction**

Einstein Platform is a way to transform a computer in a next-generation Newton N2 (MP2x00, eMate 300).

Einstein is a project to unchain NewtonOS from existing hardware. More information can be found on the Einstein Project home page: <http://kallisys.com/newton/einstein/> and on the OpenEinstein page on Google Code: <http://code.google.com/p/einstein/>

## 2 Requirements

Einstein Platform currently runs on the following machines :

- Arm-linux PDAs with X11.
- Nokia Internet tablets 770 and 880.
- Mac G4/G5 computers with 10.3.9 or higher.
- Mac Intel computers with 10.4.6 or higher.
- Windows computers with Cygwin and X11.

Einstein Platform requires about 40 MB of storage.

- Einstein Platform requires an MP2x00 US, an MP2x00 D or an eMate 300 ROM image.

Instructions about how to extract the ROM are available in the next section.

PLEASE DO NOT ASK ME FOR A ROM FILE. I will not provide you with any. NewtonOS ROM is copyright by Apple Computer, Inc and licensors.

## 3 Extraction of the ROM from your Newton

Two methods are available: via a serial line or via TCP/IP (i.e. via an Ethernet access).

### 3.1 Hammer/Newtsbug (serial line)

Using a low-level debugger such as Hammer or Newtsbug, you can make a dump of the memory. This is slow and works over the serial line.

#### Requirements:

- A computer running in Classic or MacOS < X
- Hammer or Newtsbug (they can be found on UNNA, <http://www.unna.org/>)
- A serial connection between your Newton and the Mac:
  - a built-in serial port for computers booting in MacOS < X
  - or a USB to Serial port adapter compatible with Classic.

#### Steps:

- Install Debugger Connection or Newtsbug Connection package on your Newton.
- Plug the Newton with the Mac using the serial line.
- Run Hammer or Newtsbug on the Mac.

A standard get file dialog appears: choose the debugging image corresponding to your Newton (Senior CirrusNoDebug image, Senior DCirrusNoDebug image or Newt KNoDebug image for the MP2x00 US, MP2100 or eMate 300 respectively).

- Tap the Debugger Connection or Newtsbug Connection package on your Newton and choose connect.
- On the Mac, once the connection is established, choose Save Memory from the File menu.
- Save memory between 0 and 00800000 (8 MB).

Wait.

### 3.2 ROM Dumper (TCP/IP)

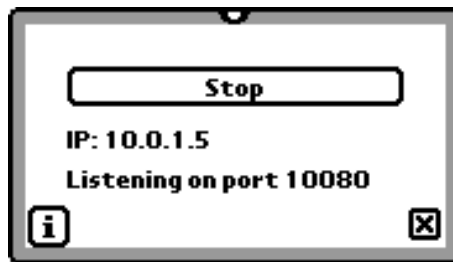
ROM Dumper is a faster approach but requires an internet connection between your Mac (or any Unix computer) and your Newton.

#### Requirements

- A working TCP/IP or Internet connection between your Newton and your Mac.

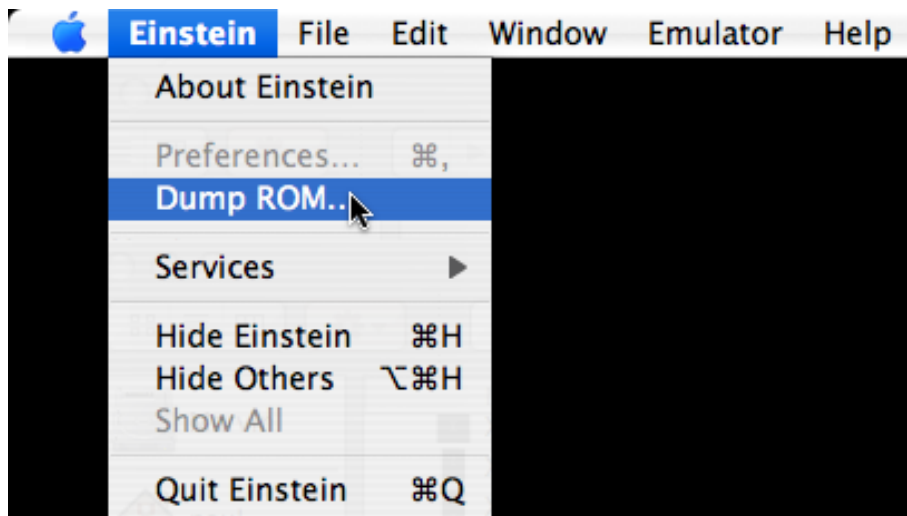
Steps:

- Install provided ROM Dumper package on your Newton.
- Tap the ROM Dumper icon in the Extras Drawer.
- Tap start.
- If your Newton isn't connected to the Internet yet, choose a connection method. It's also the time to insert your Ethernet/WiFi card.
- Note the IP of the Newton (ROM Dumper mentions it).



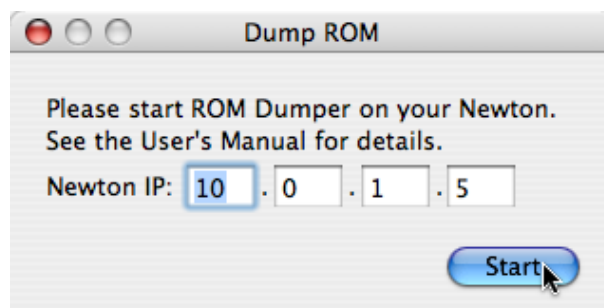
*ROM Dumper listening*

- Launch Einstein Platform (the GUI version).
- Choose Dump ROM from the Einstein menu.



*Dump ROM Menu Item*

- Type the IP address of your Newton.



*Dump ROM Panel*

- Click start.
- Mention where to save the Newton ROM. Be careful not to erase previously dumped ROM file if you are dumping the ROM of several different Newton models.
- Wait (a little bit).

The platform will be configured to use the newly dumped ROM.

Alternatively, you can use nc(1) command line tool.

## 4 Einstein on MacOS X

Just copy the GUI application to your hard drive and double click it. The first time, it will ask you to tell it where the ROM image is.

You can run Einstein Platform full screen. To exit Einstein, go to the Extras Drawer, tap the [i] button and then choose Quit Einstein.

On MacOS X, Einstein application is scriptable. You can install packages or evaluate NewtonScript code within Einstein Platform using AppleScript.

### 4.1 Using the CLI flavor on MacOS X

On desktop computers, the CLI flavor should mainly be used to access the log and/or the monitor mode. It is therefore intended for developers.

- Name the ROM dump either: 717006 (MP2x00 US) 737041 (MP2100 D) 747129 (eMate 300)
- Put the file in the data directory, next to Einstein.rex file (Einstein.rex is the ROM Extension for Einstein Platform, it includes Einstein drivers and Frank Gruendel's NewtTest program).
- Then, in the Einstein directory, launch Einstein with: `$ ./einstein --machine=XXXX data`

where XXXX should be either: 717006 (for a MP2x00 US) 737041 (for a MP2100 D) 747129 (for an eMate 300)

### 4.2 Options

`./einstein --help` will print some help about the options. The options are summarized below:

`--audio` or `-a`

Select the audio driver (null, portaudio or coreaudio). `--audio=null` will disable sound. `--audio=portaudio` will choose portaudio sound driver. Default is coreaudio.

`--width`

Set the width of the screen (in portrait mode). Default is 320.

`--height`

Set the height of the screen (in portrait mode). Default is 480.

`--log` or `-l`

Set the log file. Default is to not log. This option is incompatible with `-monitor`.

`--machine` or `-m`

Set the machine. Choose 717006 for a MP2x00 US, 737041 for a MP2100D or 747129 for an eMate 300. `--machine` option can be omitted with a 717006 ROM file.



`--monitor`

Run in monitor mode.

`--ram`

Set the RAM size in 64 KB increment. 1 will mean 64 KB of RAM. 64 is the default setting (4 MB). The maximum is 255 (nearly 16 MB).

## 5 Einstein on arm-linux PDAs

### 5.1 Quickstart on the Zaurus

I used a Zaurus SL-5500 (thanks Sylvain) with OpenZaurus (<http://www.openzaurus.org/>) 3.5.4-rc ROM and the following packages:

- libstdc++6\_3.4.3-r10\_arm.ipk
- libts-0.0-0\_0.0cvs20050403-r18\_arm.ipk
- libx11-6\_6.2.1-r2\_arm.ipk
- libxau0\_0.1.1-r1\_arm.ipk
- libxcalibrate0\_0.0cvs20050403-r0\_arm.ipk
- libxdmcp0\_0.1.3-r1\_arm.ipk
- libxext6\_0.0cvs20050222-r1\_arm.ipk
- libxfont1\_1.4.2-r2\_arm.ipk
- libxft2\_2.1.6-r1\_arm.ipk
- libxrandr2\_1.0.2-r1\_arm.ipk
- libxrender1\_0.8.4-r1\_arm.ipk
- tslib-conf\_0.0cvs20050403-r18\_collie.ipk
- ttf-bitstream-vera\_1.10-r2\_all.ipk
- xserver-kdrive-fbdev\_20050207-r0\_arm.ipk
- xtscal\_0.6.3-r0\_arm.ipk

Any more recent version of OpenZaurus and these packages should do it.

I first installed the Bootstrap Image with the zImage-64-0 to have enough RAM for Einstein Platform. Then, I copied all these packages to a CF card and I installed them all at once with the ipkg command line tool on the Zaurus (with something like `ipkg install /mnt/cf/*.ipk`).

Then, I copied the three elements from the release archive as well as the 710006 ROM image to the root of the CF card. And I launched the script with: `/mnt/cf/start-einstein.sh`

### 5.2 More details for other PDAs

Please note that Einstein Platform requires quite a large amount of RAM. Most old PDAs do not provide enough RAM for third party application unless you enter some developer mode or install a bootstrap image.

- Name the ROM dump either: 717006 (MP2x00 US) 737041 (MP2100 D) 747129 (eMate 300)

- Put the file on the compact flash, next to Einstein.rex file.
- Run the X server. Xfbdev on the Zaurus needs to be run at 270 degrees:  
`Xfbdev -screen 320x240@270 -dpi 100 &`
- Run Einstein Platform with:  
`/mnt/cf/einstein --machine=XXXX --width=YYYY --height=ZZZZ /mnt/cf`

where XXXX should be either: 717006 (for a MP2x00 US) 737041 (for a MP2100 D) 747129 (for an eMate 300)

and YYYY and ZZZZ should be set properly.

### 5.3 Options

`./einstein --help` will print some help about the options. The options are summarized below:

`--width`

Set the width of the screen (in portrait mode). Default is 320.

`--height`

Set the height of the screen (in portrait mode). Default is 480.

`--log or -l`

Set the log file. Default is to not log. This option is incompatible with `-monitor`.

`--machine or -m`

Set the machine. Choose 717006 for a MP2x00 US, 737041 for a MP2100D or 747129 for an eMate 300. `--machine` option can be omitted with a 717006 ROM file.

`--monitor`

Run in monitor mode.

`--ram`

Set the RAM size in 64 KB increment. 1 will mean 64 KB of RAM. 64 is the default setting (4 MB). The maximum is 255 (nearly 16 MB).

## 6 Developer notes

### 6.1 Compiling and porting Einstein

The documentation for obtaining the source code of Einstein, compiling it and porting it is available on Google Code web page: <http://code.google.com/p/einstein/>.

### 6.2 CLI commands

The command line interface is intended for developers and hackers.

Using the cli flavor, you will be provided with a prompt. Typing help will provide a small help about the available commands.

### 6.3 Monitor mode

The monitor mode uses a disassembler from the NetBSD project (the kernel disassembler for the arm32 port). You start in monitor mode by specifying the `--monitor` option to the command line program.

The monitor mode can be considered as an enhanced low-level debugger. The help command displays a short help for the available commands.

One of the main advantage of the monitor is that you can set breakpoints. You can also halt the emulator by calling the `|Einstein:BreakInMonitor|` global function (it doesn't take any parameter).

The following breakpoints are enabled by default:

- NewtonOS UND instructions to pass strings to the debugger (typically followed by a reboot). They are executed (i.e. the Newton will reboot), but the Newton is halted and the string is printed to the monitor.
- Some violations that shouldn't happen.

You cannot use `--log` option with `--monitor` because in monitor mode, the log is always enabled. You can save the log to a file (it scrolls on the monitor screen).

The monitor mode uses a file with symbols. This file should be named after the ROM file, e.g. 717006.symbols. The syntax is:

```
address <tab> symbol <tab> comment
```

Addresses should be sorted.

This file is very easy to generate from the debugger images that are used with Hammer and Newtsbug. Use Newton C++ Tools DumpAIF program with the `-s` option to dump the list of symbols. Then you can process all the lines with `research` and `replace` or `sed` or `awk` at your convenience. The symbols can be unmangled with `Unmangle` tool coming with MPW.

## 6.4 Logging

Quite a large amount of log lines are generated by Einstein. These are used during the development of Einstein emulator. It also helps to understand what's going on.

Starting with UP2 release, you can generate log lines from NewtonScript calling the `|Einstein:Log|` global function.

E.g.:

```
|Einstein:Log|("Hello World")
```

will display `Hello World` in the log. If log is disabled, the string will be output on stdout.

Please note that your string is converted to ISO-8859-1 before being printed, so non latin characters will not be printed properly in the log.

## 7 Relativity Developer Guide

### 7.1 Introduction

Relativity is a new technology embedded into Einstein to allow packages for Einstein to take advantage of host APIs.

Relativity can only be used through NewtonScript for the moment. To use relativity, you need to call `OpenNativeLibrary` NewtonScript function to open a library from the host system.

`OpenNativeLibrary` (*libraryName*)

*libraryName* is a string representing the name of the library to open. Valid names include "libc" or "libGL".

This function returns a library object and throws an exception if an error occurs (for example if the library cannot be found).

*libraryObject*: `GetFunction` (*functionSpecs*)

*functionSpecs* is a frame describing the function to import. This frame has three required slots :

- `name` the name of the function.
- `args` an array with the types of the arguments, from left to right. The types are symbols (see below).
- `result` the type of the result (see below).

This function returns a NewtonScript function you can call.

The types can be the following:

- `'uint8` unsigned integer, 8 bits
- `'sint8` signed integer, 8 bits
- `'uint16` unsigned integer, 16 bits
- `'sint16` signed integer, 16 bits
- `'uint32` unsigned integer, 32 bits
- `'sint32` signed integer, 32 bits
- `'uint64` unsigned integer, 64 bits (unimplemented yet)
- `'sint64` signed integer, 64 bits (unimplemented yet)
- `'float` single precision float number (unimplemented yet)
- `'double` double precision float number (unimplemented yet)
- `'longdouble` long double precision float number (unimplemented yet)

- 'string string (const char\*)
- 'iostring string for output or input/output (char\*)
- 'binary data (const void\*)
- 'iobinary data for output or input/output (void\*)
- 'pointer any pointer on data.

Some types are illegal for return or argument types (for example, you cannot use `iostring` for the return type, use `string` instead).

Then, you can call the function naturally by providing Newton data. The result is returned in the form of Newton data.

Once you are done, you can close the library with the `:Close()` method.

```
libraryObject:Close()
```

Close the library. The library object should no longer be used afterwards.

## 7.2 Sample code

In this section, we will describe how to call a simple function, `gethostname` that returns the (network) name of the host.

The manual page of `gethostname` provides the necessary information to call the routine.

```

GETHOSTNAME(3)          BSD Library Functions Manual          GETHOSTNAME(3)

NAME
  gethostname, sethostname -- get/set name of current host

LIBRARY
  Standard C Library (libc, -lc)

SYNOPSIS
  #include <unistd.h>

  int
  gethostname(char *name, size_t namelen);

```

This page shows that `gethostname` is in the `libc` library. It also provides the type of the arguments and of the result. The following code calls the function and returns the result.

```

begin
  // Open the native library.
  local libc := OpenNativeLibrary("libc");
  // Define the function.

```

```
local gethostnameFn := libc:GetFunction({
  name: "gethostname",
  args: ['iostring', 'sint32'],
  result: 'sint32});

// Allocate a buffer to store the hostname.
// This buffer is a string of 127 unicode characters (+ null).
local theBuffer := MakeBinary('string', 256);
// Call the function.
local theResult := call gethostnameFn with (theBuffer, 127);

// Close the library.
libc:Close();

// Determine the outcome of the call.
if (theResult = 0) then
  // Success.
  return theBuffer;
else
  // Failure
  return "gethostname failed";
end;
```



## 8 Known problems

- Serial ports are not emulated.
- PCMCIA cards are not emulated. The sockets aren't entirely emulated yet.
- The sound volume is reported to be settable by software but the changes are ignored (except when the sound is off on the Newton).
- Sound input isn't emulated.
- An error occurs when trying to install system patches.
- Keys are not translated properly except on the Mac (the keyboard is useless).
- Rotation doesn't always work well.
- It is awfully slow on ARM PDAs.

Please report any problem you experienced that is not on this list on the Google Code page (<http://code.google.com/p/einstein>).

## 9 Changes History

9/07/07 Einstein Platform 2007.7

- New version of the emulator module, which uses less RAM and is PC-agnostic.
- Fixed the emulation of several instructions (and wrote regression tests).
- Compilation for Nokia OS 2006.
- Experimental direct ARM to ARM translation.
- Now open source!
- Ported to Cygwin [matthiasm]

5/31/06 Einstein Platform 2006.6

- Features:
  - Initial release of Relativity for Einstein.
  - Accelerated some transfers between Einstein and Host.
  - Included several keyboard mappings.
  - Update of PortAudio (from CVS).
  - Included a new set of icons by Michael Vacík. The toolbar icons reflect the state of Einstein.
  - Fixed an endianness problem with the Cocoa screen manager on Intel machines.
  - The state is now shown on the screen (instead of just: screen is off).
  - Improved the Cocoa preferences panel.
  - Included AppleScript support to evaluate newton script code and to install packages.
- Bug fixes:
  - Fixed a bad bug in the memory emulation that caused crashes, especially with the JIT page cache.
  - Fixed a bug yielding to an abort when the platform was quitted.
  - Fixed a bug with CoreAudio sound on Intel machine.
  - PortAudio sound driver now reverts samples an little endian targets.
  - The tablet region was incorrectly limited to 1023x1023. It's now set to NewtonOS size (2047x2047).

1/14/06 Einstein Platform Beta

- Initial release.

## 10 License

Einstein Platform is copyright 2003-2007 by Paul Guyot. It is available under the GNU General Public License v2.

GNU GENERAL PUBLIC LICENSE  
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.,  
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA  
Everyone is permitted to copy and distribute verbatim copies  
of this license document, but changing it is not allowed.

### Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original

authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

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

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's 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 give any other recipients of the Program a copy of this License along with the Program.

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 Program or any portion of it, thus forming a work based on the Program, 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) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third

parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, 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 Program, 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 Program.

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

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a) 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; or,
- b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, 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 executable. However, as a special exception, the source code 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.

If distribution of executable or 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 counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program 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.

5. 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 Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program 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 to this License.

7. 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 Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program 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 Program.

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.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program 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.

9. The Free Software Foundation may publish revised and/or new versions of the 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 Program 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 Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, 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

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "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 PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. 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 PROGRAM 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 PROGRAM (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 PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

#### How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
<one line to give the program's name and a brief idea of what it does.>  
Copyright (C) <year> <name of author>
```

```
This program is free software; you can redistribute it and/or modify  
it under the terms of the GNU General Public License as published by  
the Free Software Foundation; either version 2 of the License, or  
(at your option) any later version.
```

```
This program is distributed in the hope that it will be useful,  
but WITHOUT ANY WARRANTY; without even the implied warranty of  
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the  
GNU General Public License for more details.
```

```
You should have received a copy of the GNU General Public License along  
with this program; if not, write to the Free Software Foundation, Inc.,  
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.
```

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this



when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright interest in the program
`Gnomovision' (which makes passes at compilers) written by James Hacker.
```

```
<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice
```

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License.

**Einstein Platform includes portaudio v19 (from CVS).**

```
/*
 * PortAudio Portable Real-Time Audio Library
 * Latest Version at: http://www.portaudio.com/
 *
 * Copyright (c) 1999-2000 Phil Burk and Ross Bencina
 *
 * Permission is hereby granted, free of charge, to any person obtaining
 * a copy of this software and associated documentation files
 * (the "Software"), to deal in the Software without restriction,
 * including without limitation the rights to use, copy, modify, merge,
 * publish, distribute, sublicense, and/or sell copies of the Software,
 * and to permit persons to whom the Software is furnished to do so,
 * subject to the following conditions:
 *
 * The above copyright notice and this permission notice shall be
 * included in all copies or substantial portions of the Software.
 *
 * Any person wishing to distribute modifications to the Software is
 * requested to send the modifications to the original developer so that
 * they can be incorporated into the canonical version.
 *
 * THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
```

```
* EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
* MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
* IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR
* ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF
* CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
* WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
*
*/
```

Einstein Platform also includes NetBSD arm disassembler.

```
// $NetBSD: disassem.c,v 1.14 2003/03/27 16:58:36 mycroft Exp $
//
// Copyright (c) 1996 Mark Brinicombe.
// Copyright (c) 1996 Brini.
//
// All rights reserved.
//
// Redistribution and use in source and binary forms, with or without
// modification, are permitted provided that the following conditions
// are met:
// 1. Redistributions of source code must retain the above copyright
//    notice, this list of conditions and the following disclaimer.
// 2. Redistributions in binary form must reproduce the above copyright
//    notice, this list of conditions and the following disclaimer in the
//    documentation and/or other materials provided with the distribution.
// 3. All advertising materials mentioning features or use of this software
//    must display the following acknowledgement:
// This product includes software developed by Brini.
// 4. The name of the company nor the name of the author may be used to
//    endorse or promote products derived from this software without specific
//    prior written permission.
//
// THIS SOFTWARE IS PROVIDED BY BRINI ``AS IS'' AND ANY EXPRESS OR IMPLIED
// WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
// MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
// IN NO EVENT SHALL BRINI OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT,
// INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
// (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
// SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
// HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
// STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN
// ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
// POSSIBILITY OF SUCH DAMAGE.
```