

# Einstein Platform User's Manual

For Einstein Platform 2006.6

May 31, 2006

# 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 (with SoftFPU) 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>Einstein on the Nokia 770</b>	<b>12</b>
<b>7</b>	<b>Developer notes</b>	<b>13</b>
7.1	CLI commands . . . . .	13
7.2	Monitor mode . . . . .	13
7.3	Logging . . . . .	13
<b>8</b>	<b>Relativity Developer Guide</b>	<b>15</b>
8.1	Introduction . . . . .	15
8.2	Sample code . . . . .	16
<b>9</b>	<b>Known problems</b>	<b>18</b>
<b>10</b>	<b>Changes History</b>	<b>19</b>
<b>11</b>	<b>License</b>	<b>20</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/>

## 2 Requirements

Einstein Platform runs on the following machines :

- Arm-linux PDAs with X11, libc 2.3.3 with soft FPU and enough memory, such as the Zaurus SL-5500 with OpenZaurus.
- A Nokia 770.
- A Mac with a G4 or higher, running 10.3.9 or higher.
- A MacIntel running 10.4.6 or higher.

Einstein Platform requires 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)

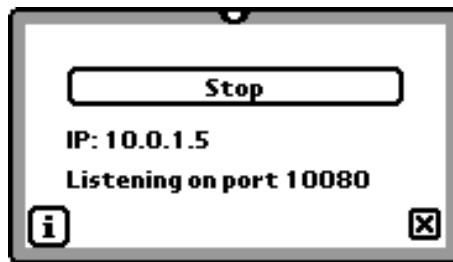
ROM Dumper is a faster approach but requires an internet connection between your Mac (or any Unix computer) and your Newton.

##### Requirements

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

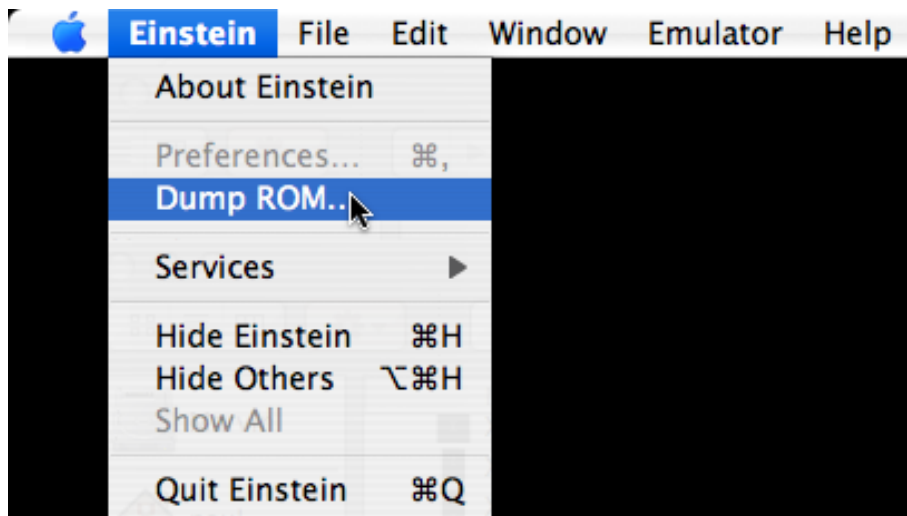
Steps:

- 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).



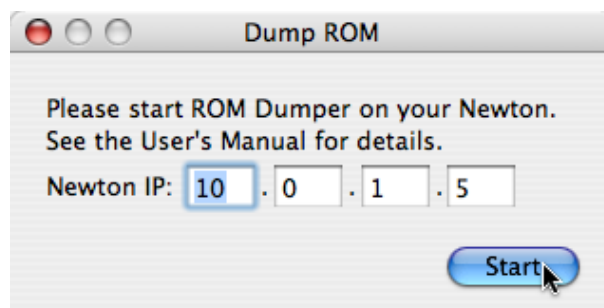
*ROM Dumper listening*

- 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 (with SoftFPU) 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`).

I fixed the date with `date 053112002006`.

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

I haven't tested Einstein Platform on other arm-linux PDAs. Please note that Einstein Platform requires `libc 2.3.3` and quite a large amount of RAM. Most 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 Einstein on the Nokia 770**

The first step is to enable root mode. Connect the Nokia to a desktop computer and use the flasher developer tool to enable root mode.

## 7 Developer notes

### 7.1 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.

### 7.2 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.

### 7.3 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.

## 8 Relativity Developer Guide

### 8.1 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.

## 8.2 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;
```

## 9 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.

Please drop me a line (<mailto:pguyot@kallisys.net>) if you experience a problem which is not in this list.

## 10 Changes History

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.

## 11 License

Einstein Platform is copyright 2005-2006 by Paul Guyot. It is provided as is, without any warranty of any kind.

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.
```