

PainForm — An overview

J.M. van Schalkwyk

February 27, 2009

Contents

1	What is PainForm?	3
2	Why use PainForm? Why not?	4
3	About the documentation	5
3.1	Fundamental documents	5
3.2	PDA program	5
3.3	Ancillary programs	6
3.4	Re-creating the PDF documentation	6
4	Installation guidelines	7
4.0.1	Hardware	7
4.0.2	Program prerequisites	7
4.1	Full installation	8
5	Detailed Installation	10
5.1	Installing Cygwin, GCC and make	10
5.2	Installing Pilrc and PRC-tools	11
5.3	Installing the Palm Development kit	11
5.4	Using PainForm files	12
5.5	The <i>painform</i> directory	13
5.6	DogWagger 2.1	14
5.7	Installing required programs	14
5.7.1	Installing and using AutoIt	14
5.7.2	Installing and using PFB	14
5.7.3	Installing and using filepc2pda	15
5.7.4	Conflict with HotSynch	15

<i>CONTENTS</i>	2
6 Modifying PainForm	16
7 Synchronisation with other databases	16
8 Debugging PainForm	17
8.1 Using POSE	17
8.2 Using GNU debug	18
8.3 Profiling	20
8.3.1 Profiling from within code	20
9 Future Developments	21
10 The GNU Public Licence, V2, June 1991	22
10.1 Preamble	22
10.2 Terms and Conditions	23

1 What is PainForm?

PainForm is a PDA-based data-entry program which captures analgesia-related information for a hospital pain service. This version is a very early one (V 0.95) but it has been released into the public domain to allow other people to tinker with it, under the terms of the GNU Public Licence (Section 10). PainForm has several interesting features:

- PainForm is carried on a PDA (personal digital assistant) and data are entered at the bedside. The PDA is later synchronised with a desktop database which closely follows the structure and function of the PDA-based program. Data flow between PC and PDA in both directions. At present PainForm runs on PDAs based on the Palm operating system (PalmOS). Multiple PDAs can synchronise with the desktop database.
- The menu system used is *simple*, flexible and intuitive, and the same whether the PDA or desktop program is run. We avoid busy screens.
- Most items and item lists are customisable according to user preference, including ward numbers and names and drug details.
- PainForm is freely available under the GNU Public Licence (GPL), a copy of which is available below — source code will always be freely available.
- It tries hard to provide a substrate for *cross-platform* implementation. The desktop version uses Perl, dialects of which work on almost any operating system. We've avoided platform-specific Perl as far as we can, but there is one catch. We use Perl 5.6 (and can see no pressing reason to upgrade) but recent versions of Perl lack an easily installed version of Tk, which we use for the graphical user interface on the desktop. Either use the old version, or tinker a bit.¹
- It is based on SQL, the international standard for database implementation. It uses 'ODBC', another standard which allows the program to talk to most databases which comply with the ODBC standard.² SQL is used both in the desktop database, and on the PDA in a scaled-down but powerful form.
- Program behaviour is facilitated using a powerful scripting language which runs in the same fashion on both PDA and desktop PC.

¹Some '5.6' versions available on the web now also lack Perl/Tk, so be careful!

²Otherwise known as the SQL-CLI or SQL Call-level interface.

- The menu system used by PainForm is completely represented in the database, allowing easy customisation of menus, provided you know what you are doing.
- Source code and documentation are provided together, so that both full PDF documentation and Perl / C / C++ source code can be obtained from the same (.TEX) source files. Documentation and code are synchronised.

2 Why use PainForm? Why not?

Although there are apparently compelling reasons to use PainForm — it's freely available under the GPL, with source code, and it's small and orthogonal — we have not provided a version that you can simply download and run. We have provided the code largely so that those in the know can tinker with it. Even if you're fluent in C, C++, Perl and SQL you are still likely to have the odd hiccup, and there are areas where the code is very crude.

This suite of programs is under continual development, but no warranty of fitness is provided for any use of the program. Please see the GPL (Section 10) for further details.

In order to modify the program (which is necessary for all but the most trivial of applications) the user will probably need to be familiar with SQL, although the PC-based menu system can be used to make trivial modifications, for example entry of new personnel.

In order to substantially modify the program (change the coding, further develop it, or implement it under e.g. Linux) the following skills are required:

1. A good knowledge of SQL;
2. A substantial knowledge of Perl (needed to modify desktop application);
3. A good working knowledge of C and C++ as it applies to the PalmOS PDA.

If you are running the program in the United States of America, be aware that we have *not* implemented encryption, and the program in its current form is probably therefore not HIPAA compliant.

Our thoughts about this is that on the PDA, the most robust protection of the data is related to the user carrying it around on their person; in addition, small amounts of data are carried, of minimal or no use to the average intruder.

3 About the documentation

The PainForm documentation is all written in \LaTeX , Donald Knuth's powerful document formatting software. We have converted the original \LaTeX into PDF documentation, but using our DogWagger program, the same \LaTeX files can be used to generate complete source code for all Perl, C and C++ programs which run on the PDA and PC.

The only apology we will make for the entire project is our inability to make really pretty documentation. Our \LaTeX skills are rudimentary.

Documentation is divided into three sections.

3.1 Fundamental documents

These documents, which provide the basis for all other documentation, are as follows:

1. PainFormReadme.tex — this document;
2. PainForm.tex (and PainForm_105.tex) — an associated (backup) paper form;
3. PDAdata.tex — overview of data display on the PDA;
4. AnalgesiaDBpart1.tex — the basic database structure, including data tables;
5. AnalgesiaDB2.tex — the PDA menu structure, with screenshots and scripting;
6. PerlPgm.tex — the complete Perl/tk program which runs on the desktop, including interfacing to other databases;

3.2 PDA program

The PDA program is written in C++, apart from the several libraries, which are written in C. The relevant documents are:

1. CProgMain.tex — the main C++ program. The following are all libraries:
2. ScriptingLib.tex — A library used to interpret scripting commands.
3. ErrLib.tex — Used for error display;
4. Sql3Lib.tex — Library for processing of SQL on the PDA;

5. NumericLib.tex — Rudimentary numeric routines which will ultimately be IEEE754r compliant.
6. CacheLib.tex — used in database caching, vital for reasonably speedy SQL interpretation on the PDA.
7. IdxLib.tex — Experimental library to create indexes into the PDA SQL databases.

Note that for many of the smaller components, especially the libraries, the .TEX document is simply a wrapper for the C files.

3.3 Ancillary programs

The following are extremely useful, but not core programs:

1. ConsoleLib.tex — allows us to write to a ‘console’ for debugging and other purposes;
2. OsBox.tex — simple viewing of the console;
3. DogWagger21.tex — generation of Perl, C and C++ source code from .TEX files.

3.4 Re-creating the PDF documentation

The source files for PainForm are also documentation files! The .TEX files can all be converted to PDF files using the program PdfLatex. The best way to do this is from a good L^AT_EX editor such as WinEdt. You will of course need L^AT_EX itself — go for *MikTeX*.

4 Installation guidelines

4.0.1 Hardware

The following hardware is required:

1. A PalmOS-based PDA running version 4.0 or greater of PalmOS. We have successfully used both an elderly Zire 31, and Treo 650s and 680s from Palm/PalmOne. The program should work on any recent PalmOS-based PDA. The PDA will need a USB cable to communicate with a PC.
2. A PC running Windows (Because of the way we have designed the program, using Perl and ODBC, porting PainForm to other operating systems is feasible, but we haven't yet done this. The Perl code should run almost unchanged under e.g. Linux, but communication with the PDA will have to be reshaped using e.g. ColdSync; it's actually more difficult to achieve adequate communication between the PDA and Windows-based PCs than using other systems such as Linux, owing to the crippling liabilities of Windows).

4.0.2 Program prerequisites

The following freeware or similar programs are required in order to run the PainForm suite:

1. ActivePerl version 5.6, obtainable off the Internet. Apart from the standard installation, the packages `File::stat`; and `Time::localtime` are needed (but should be available in any recent version); more taxing is the requirement to install `Tk::WaitBox`; and `Tk::ProgressBar`. If you fail to install these packages, this might be a signal that you need assistance from a propeller-head; if you really struggle with `WaitBox/ProgressBar` you might even hunt down the relevant lines in the Perl program and comment them out!
2. An ODBC-compliant database. We have used Ocelot SQL because of its availability, speed, friendliness, and SQL compliance; modification of SQL code might conceivably be required if other proprietary databases are used. A reasonable alternative (now that Ocelot is no longer being developed) might be MySQL.
3. The small PDA program *filepc2pda*, to move data between PDA and PC (The native PalmOS `hotsynch` sucks, and we were disinclined to lock into Microsoft-based conduits, one of the several design flaws of PalmOS). The program was written by Mathias Lütke.³

³Obtainable from <http://www.ghisler.com/serial.htm>

4. The Windows program Palm File Browser (PFB)⁴ You will probably need to register with MyTreo. The program is also available on the Internet without registration requirements, if you look around. The executable is called PFB.exe. This program helps circumvent the limitations of hotsynch, and obviates the need for a conduit. You will also need the files *sertransplg.dll* (which came in the zip with PFB), and the elusive file *USBPort.dll*. The latter file should have come with your Palm software; otherwise get a recent version off the 'Net.⁵
5. The program AutoIt⁶ This program is again needed to overcome the crippling limitations of Windows!⁷

4.1 Full installation

If you wish to extensively modify the PainForm program, the most important set of programs you will need to install is the PalmOS SDK. In order to use these programs, you will need GCC (the GNU C/C++ compiler), which runs under UNIX. Under Windows, use CYGWIN, a UNIX equivalent for Windows. The necessary source code is all provided with the basic installation above (!) but you will need to create the relevant subdirectories in the *PalmDev* directory in the root of the same drive on which you installed the basic kit.

Full installation of PainForm will allow you to modify the program. Apart from the skills listed above in Section 2, you will need:

1. **Cygwin**. This is a Windows-based version of UNIX. Even better, if you can, work on a UNIX or Linux system.
2. **PRC-tools** — a freeware development environment for Palm OS. When you get this, get PilRC too. This “resource compiler” is used to ‘make binary resource files from a resource script file’. Our use of such resource files will be minimal, but it’s still appropriate to get these tools.
3. **A Palm OS SDK** (Software development kit) You should go for version 4.0 or greater.
4. The GNU C++ compiler, GCC.

⁴Try <http://mytreo.net/downloads/details-220.html>

⁵There may be a copy elsewhere, e.g. under Win2K in the directory */WINNT/system32*.

⁶Available from e.g. <http://www.autoitscript.com/autoit3/docs/>

⁷We will eventually provide a stand-alone executable to fulfil the required functions but in the meantime go get AutoIt, an absolutely brilliant program.

5. POSE (from palmos.com). POSE requires ‘skins’ and a ROM image. Skins are pictures of devices, and a ROM image is the ‘operating system’ (not as freely available as the other components, but you can download an image from your ‘Pilot’, or obtain a generic image from PalmOS).
6. A good text editor which allows easy editing of text and \LaTeX (.TEX) files. We recommend WinEdt — it’s the only software we’d recommend you pay good money for, and it’s not expensive.
7. MikTeX, needed to process \LaTeX documentation files.
8. Our Perl program *DogWagger* version 2.0 or greater.

Don’t despair when reading the above list! The programs are all readily available, and in the following section we go through installation of each component in more detail. A lot of spadework is required, but if you have the time, it’s not too taxing, and it’s free.

5 Detailed Installation

In this section we look at each component of PainForm, and how it's installed. We assume you will be eating the whole enchilada, including re-creating the PRC files from scratch! Our documentation is for MS Windows; smarter users of other operating systems will need to modify their approach.

5.1 Installing Cygwin, GCC and make

Under Microsoft Windows, go through the following steps:

1. Create a Cygwin directory, for example `c:\cygwin2`.
2. Make a downloads subdirectory: `c:\cygwin2\downloads`.
3. Download the Cygwin setup file from [Cygwin](#) into the above *downloads* subdirectory.
4. Run the Cygwin program **setup.exe**
5. In the 'Choose a download source' menu, choose 'Install from Internet'
6. As root directory, choose `c:\cygwin2`
7. As local package directory, choose `c:\cygwin2\downloads`
8. Choose one of the download sites.
9. when you get to the *Select Packages* menu click on the 'View' button until the text to the right of this button reads 'Full'. You can now scroll down, looking through the package names.
10. See that by default the standard packages are selected (look at the 'New' column on the left).
11. See how the packages **gcc** and **make** are *not* yet selected. Click on the entry on the left (in the 'New' column) to select these packages too. This is most important!
12. Wait while Cygwin is downloaded. This may take some time. Cygwin will tell you when it has finished.

5.2 Installing Pilrc and PRC-tools

Installation is similar to the above:

1. Run **setup.exe** once more, to further download from the Internet. Under 'Choose a download site' in the 'User URL' box type in the following:

```
http://prc-tools.sourceforge.net/install
```

2. Click on 'Add' and then 'Next', and select the packages **prc-tools** and **pilrc** as you did for GCC and make. Install these packages.

5.3 Installing the Palm Development kit

If you're lucky enough to have the smaller (and functional) Version 4 SDK, no longer available from the PalmOS site, go through the following steps:

1. In DOS make a special palm directory, eg on the C: drive say:

```
md \PalmDev
```

2. Run Cygwin, and make a UNIX directory:

```
mkdir /PalmDev
```

3. From within Cygwin, symbolically link the UNIX and DOS directories!
Thus:

```
mount -f "C:\PalmDev" /PalmDev
```

You only have to do this once.⁸

4. Install the PalmOS SDK. Unzip the whole file (recreating the sub-directories) into, in our example, C:\PalmDev.

Note that there is an associated readme file. Read it! It is possible that you may have to play around, renaming various directories. The important thing is that in your PalmDev directory (now accessible from both UNIX and DOS!) you will have a directory called `sdk-4`. This should in turn have two important subdirectories, called `lib` and `include`.⁹

⁸The corresponding command to unmount (remove) this symbolic link is *umount*. Note that if you are using a drive other than C: you'll have to alter the mount command appropriately.

⁹If you can't find the `sdk-4` directory, then look at the directories you do have — you may have unzipped it into one of these, and have to move it up a level!

5. Run Cygwin. Then change to the /PalmDev directory,¹⁰ and then type in the following:

```
ln -s sdk-4 sdk
```

It looks tricky, but all it does is to create a ‘soft’ link between a new directory (sdk, just made!) and the sdk-4 directory. When you use GCC, it will see this ‘sdk’ directory as the one it needs to use.¹¹

6. Finally (whew!), still in Cygwin, type in:

```
palmdev-prep
```

What this should do is prepare links etc so that GCC (and friends) will see the files they need to see. Otherwise you might have to try and create a whole lot of such symbolic links and whatnot — a real pain with previous editions.

5.4 Using PainForm files

We here describe the directories required by PainForm in creating and modifying PRC files.

PalmDev The main development directory;

PalmDev/testing The directory used by PainForm. This contains the source files required to create the file *pain5.prc*;

PalmDev/testing/err For error library files.

PalmDev/testing/scripting For scripting library files.

PalmDev/testing/numeric For numeric library files (rudimentary at present).

PalmDev/testing/sql3 For SQL library files

PalmDev/testing/idx For the experimental indexing library.

PalmDev/testing/cache For the important SQL caching library.

¹⁰cd /PalmDev

¹¹To remove this link, simply say *rm sdk*.

PalmDev/testing/console For the console library.

PalmDev/testing/osbox For the primitive ‘osbox’ program, which displays information written to the console, and stored in the PalmOS database *CONSOLE.PDB*.

5.5 The *painform* directory

This directory contains all the files needed to run PainForm, including MS DOS batch files, and other executables. The directories required are:

- `\painform` The main directory, containing major PainForm files including batch files, *dogwagger20.pl*, and the most important Perl file of all, *pain2.pl*. Another file, *CycMatch.pl* can be used in debugging memory leaks dumped from the PDA.
- `\painform\images` The image directory, used in creation of PDF documentation from .TEX source files.
- `\painform\data` Used to contain .SQL files, which are scripts used to generate the database for the first time.
- `\painform\csv` Used to contain .CSV files, which are database files employed in populating the original database. These files have a very specific structure, but can be generated from carefully constructed Excel data files (for example, they may contain names of staff members to be inserted into the database).
- `\painform\import` Data files imported *from* the PDA are stored in this directory. They are then processed, and new data obtained from them are inserted into the PC-based database.
- `\painform\export` Following import of data from the PDA, new export PDB files are created in this directory. They will then be moved to the PDA.
- `\painform\prc` Contains the default .PRC files to be used on the PDA. If you alter the PRC files, you *must* move the new PRC files to this directory (overwriting the old ones) if you want the new file(s) to be installed on the PDA. The Administration menu button ‘Install to PDA’ reads PRC files from here and nowhere else!
- `\painform\idx` and the subdirectories *numeric*, *console*, *err*, *sql3*, *scripting*, *osbox* and *cache* are used to retain C and C++ source files generated from .TEX files before they are exported to the *C:\PalmDev* directory.¹²

¹²This approach is clumsy, and might be replaced by direct writes to the PalmDev directory.

5.6 DogWagger 2.1

This Perl 5.6 program permits extraction of source files from .TEX files. It is used throughout our project.

5.7 Installing required programs

Due largely to the clunky nature of Windows combined with our desire to bypass the unfortunate ‘requirement’ for *conduits* imposed by PalmOS, we require use of several ancillary programs. These are:

The PDA program *filepc2pda*. We use this program on the PDA to bypass the usual PalmOS synchronisation mechanism. It talks via the USB port.

AutoIt A superb program for automating tasks in Windows, far more powerful than native or other Windows scripting languages. We use AutoIt to drive the remaining programs listed below.

PFB, the Windows *Palm File Browser*. This is the windows-side program corresponding to *filepc2pda*.

5.7.1 Installing and using AutoIt

Download AutoIt.¹³ Installation is straightforward.

5.7.2 Installing and using PFB

This is a little bit of a rigmarole:

1. Obtain the PFB program, as noted above.
2. Check that you have the file *sertransplg.dll* (which came in the zip with PFB). Put this file in the *painform* directory.
3. Find the file *USBPort.dll*. This file should have come with your Palm software; otherwise get a recent version off the ‘Net.’¹⁴ Copy this file to the *painform* directory.

That’s it.

¹³Available from e.g. <http://www.autoitscript.com/autoit3/docs/>

¹⁴There may be a copy elsewhere, e.g. under Win2K in the directory */WINNT/system32*.

5.7.3 Installing and using filepc2pda

Installation is straightforward. To install the program, download it from the Internet.¹⁵ This is the only file you need to install on your PDA using the standard PalmOS Hotsynch. Run the PalmOne program 'Palm Desktop' and select the relevant user. (Make a user called 'PainForm', for example). Click on 'Quick Install' on the Palm Desktop, and then 'Add', selecting the path of the file you downloaded, and then clicking on *filepc2pda*. Hotsynch your Palm device, and the file should be synchronised.

You will probably wish to create a new category on the palm (We'll call our new category 'PainForm'). Do so by clicking on the top right corner of the main applications screen on the device, and selecting 'Edit Categories'. Click on 'New' and enter the category name 'Level8', and then 'Ok'. Back on the main screen, click on the top left corner of the main menu and select category. You can easily change the category of filepc2pda to 'Level8'.

Connect the PC and the PDA using a USB cable. Run filepc2pda on the PDA. Ignore the initial error. Click on 'Port', and despite the fact that you are using a USB cable, do *not* click on 'USB'! Instead click on a port speed of 115200. Click on 'OK' and then 'Connect'.

Now run the *pain2.pl* program on the desktop (It is of course in the *painform* directory). Click on 'Administration' and 'Install to PDA'. If you've followed all of the above instructions, then the whole PainForm application should be moved to the PDA.

5.7.4 Conflict with HotSynch

This should not occur, unless you try to run HotSynch at the same time as you run our synchronisation. Don't do so! If you believe conflict is occurring, rather just exit the HotSynch program, but do *not* uninstall HotSynch.

¹⁵Obtainable from <http://www.ghisler.com/serial.htm>

6 Modifying PainForm

Remember that PainForm is available under the GPL, so, among other requirements, if you modify and distribute the program, you *must* distribute the source code.

7 Synchronisation with other databases

The standard version of PainForm has been written to interrogate an external database (the IDAS/SaferSleep Anaesthesia database) but depending on your requirements you may need to alter the program so that it can communicate with your local databases.

You will then need to:

1. Modify connection strings
2. Rewrite SQL (and possibly even Perl) scripting to talk to your local database.

8 Debugging PainForm

8.1 Using POSE

If you are modifying the PDA program, it is likely you will spend a lot of time resetting your PDA if you move untested programs to the PDA. Rather use POSE.

1. In a convenient directory, create a subdirectory called `pose`.
2. Unzip the POSE emulator into this directory.
3. Also obtain skins and ROM images, and keep them in appropriately named directories.
4. Run the emulator.exe file, and with a bit of fiddling you will be able to load your PRC and PDB files onto a fully functional emulation of the Palm of your choice! (Right click on the picture of the device to get the 'Install application' option).

It is a pain to move all of the PDB and PRC files required for PainForm to the PDA manually. There are several options, including writing an AutoIt script, but here's a better way.

Under MS Windows (2000 or XP) it is possible to 'HotSync' the POSE emulator to a local directory without using a null modem cable, or even being connected to a network.¹⁶ Run through the following laborious process once, and keep a copy of the POSE configuration:¹⁷

1. Run the Palm Hotsync manager, so that its icon appears on the Windows taskbar. Right click on this icon, and select 'Network'.¹⁸
2. Run POSE, right click on the emulator, choose Settings/Properties, and then ensure that the "Redirect NetLib calls to host TCP/IP" box is checked. Click OK.
3. Still in POSE, click on the Hotsync icon, followed by a click on the title at the top to activate the 'Options' menu. You will need to do some configuring:

¹⁶Depending on the level of paranoia in your system, you may need to be the, or be friends with your, network administrator to get this to work. Firewall software may also get in the way.

¹⁷Run POSE, right click, and 'Save as'.

¹⁸You may have to first create a user within the PalmOS desktop, and then click in a similar fashion on the Hotsync manager, select 'Setup' and then check the relevant box in the 'Network' tab.

- (a) First, choose “Modem Sync Prefs” and click on the Network button, followed by OK;
 - (b) Next, select the “LANSync Prefs” menu option, and choose Local HotSync and OK;
 - (c) Finally in this section, choose “Primary PC Setup” where you will need to enter the ‘network’ name of your PC,¹⁹ followed by typing in localhost as the “Primary PC Address”. Leave the subnet mask blank.
4. Your next POSE task (remember you only have to do all of this once!) is to click on “Modem” in the main Hotsync screen, and below the Hotsync logo click on ‘Select Service’. Enter the service as POSE, ‘Tap to Enter Phone’ and type in a number of zero, and click OK. Don’t fuss about the User name, password or connection. Click ‘done’.

Save this copy of POSE, and when you click on the modem hotsync logo, with luck you should be able to hsync to a local PC directory! You’ll find the PDB (and other) files stored in a local directory such as
 Program Files\Handspring\Level8\Backup
 ...depending on your installation of the PalmOS desktop, and the user name you’ve chosen (here Level8).²⁰

8.2 Using GNU debug

The command-line debugger available with GCC is immensely powerful in debugging faulty programs! You use it together with POSE. In your Makefile for the program you wish to debug (here pain5) specify: ²¹

```
m68k-palmos-gcc -g -O2 -fno-exceptions -fno-rtti
                pain5.cpp -o pain5
```

We’ve broken the single line into two for convenient reading. Now do the following:

1. Load the COFF file the above generates (as *PAIN5*) into GDB using:

```
m68k-palmos-gdb pain5
```

¹⁹You can obtain this by right clicking on ‘My Computer’ on your PC desktop, choosing ‘Network Identification’, and then looking at the name under ‘Properties’.

²⁰There are many ways to find where the files have been stored, but the easiest is to right click on the taskbar Hotsynch icon, and view the Log after synchronising.

²¹There’s a good introduction on the web by [Warren Young](#)

2. Run POSE and load *but do not run* the program to be debugged (Here `pain5.prc`);
3. Within CYGWIN type in
`target pilot localhost:6414`
(just so)!
4. Run the POSE program, and type in `cont` in the GDB window to continue until something horrible happens (best done using a debug ROM with POSE). At this point, control will go back to GDB, and you can do smart things like ...
5. Type in the backtrace command: `bt ...` to allow you to see the call stack, that is, where the offending command crashed the Palm. Wonderful, and now you can nip off and read the GDB manual for all the other smart things you can do (or simply type in `help` at the GDB command line). Type `q` to quit.

Other useful commands are:

- `break PilotMain` (set breakpoint)
- `cont`
- `bt` (stack backtrace)
- `s[tep]` (into `fx`)
- `n[ext]` (but call `fx`)
- `p[rint]` (show contents of variable, or compute expression)
- `h[elp]`
- `q[uit]`
- `l[ist]`
- `enable display`
- `p[ty]pe`

Read the documentation!

8.3 Profiling

POSE (in concert with GDB) has a whole array of profiling features. Use the Windows POSE executable with ‘profile’ in its name. Run this application (e.g. *Emulator_Profile.exe*), right click, and select Profile:Start. When you save profiling results, they are written to a text file (and a Metrowerks mwp file) in the same directory as the profiler executable.

C++ mangles names so the results can be a mess. Unfortunately owing to the current size of our main program, specifying the GCC flag *-mdebug-labels* results in compiler errors due to the increased size of the program with embedded labels. We can still use this flag with most libraries to good effect.

The tab-delimited .TXT file can be imported into Excel with minimal fuss. You can then easily browse through (and sort) the columns of data provided for all functions. Look particularly at the ‘only msec’ and ‘plus kids msec’ for how much time is expended in each function. The ‘count’ column is the number of times each function was invoked during your profiling session. Remember that you should regard all times as relative, as emulator speeds are often slower than on a good PDA, unless you have a really fast desktop!

Sorting by ‘plus kids msec’ gives a good idea of how much time is spent in each routine — inclusive of all called routines. You can create a tree of who is calling whom using the ‘parent’ and ‘index’ columns.

8.3.1 Profiling from within code

In our scripting language, we provide a command which retrieves the internal timer count as an integer. It’s called TICKS, and by using it before and after an SQL statement (for example) you can determine how many ‘ticks’ the SQL took on the PDA.²²

²²At present, this function always returns zero in Perl, as we don’t use the Perl micro-timer.

9 Future Developments

PainForm in its current incarnation is not easily configurable by the average person. We had bold intentions of writing further layers, notably one that allows easier scripting, but this is unlikely to happen soon. Over the past several years I've changed my programming philosophy considerably, and were I to rewrite this program, I would probably change to the "Test a little, code a little" paradigm for the whole project.

I've also become even more acutely aware of the unreliability, instability and insecurity of the underlying operating systems, particularly Windows, and of the dangers of current approaches to diagnostic coding and information transfer in Medicine.

You should thus regard PainForm as an experimental system with several limitations, that is unlikely to progress rapidly in the next several years. A further problem is that Palm lost the initiative several years ago, and is unlikely ever to regain it.

Look out for Linux-based PDAs as a possible future solution to the obsolescence of Palm, and the instability of Windows-based systems. Owing to the complex architecture of Windows, it will be a miracle if uncommon errors and paradoxical interactions are ever removed from this operating system.

10 The GNU Public Licence, V2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc. 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

10.1 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 Library 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.

10.2 TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

1. 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.

2. 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.

3. 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.

4. 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.
5. 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.
 6. 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.
 7. 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.

8. 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.

9. 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.
10. 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.

11. 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

12. 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.
13. 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)