

# UNIX Basics

by Peter Collinson, Hillside Systems



## Palmed Off, Not

As I write, summer is silently slipping away. For several years on this side of the big pond, autumn has approached in a kindly manner, with warm days and sunny skies. The folklore is that the mildness is due to global warming. This may be true, but it does provide a gentle change from summer to winter, which is fine in my book. Anyway, the end of the summer also saw the demise of my Psion 3a handheld. The ribbon cable in the hinged lid flexed every time it was opened, and the cable seemed to have cracked causing a short. The screen now displays an overlay of a set of black parallel lines that inhibit its normal operation. Somewhere behind these lines the system is visible and is working. The lines disappear when something is pushed against the cable (such as the plastic toothpick from my smallest Swiss Army knife) but this is not the way it was intended to work. I suspect the lines behave like the light in the fridge, they go away when the lid is closed.

I've had the Psion personal digital

assistant (PDA) for a considerable number of years and it's done sterling service. It runs for four or five months on two AA batteries—a feature that's hard to beat. It is carried my diary and address book around the world. Its little keyboard is also just about usable. I recall writing an article on it in an airport lounge on one occasion. I had also bought a time-logging application that I used to record billable work I did for various clients. I find the ability to log time is invaluable, not only because I always forget what I did when, but I can capture the odd 10 minutes from here and there, adding up to some number of hours over the month.

Getting stuff in and out of the Psion always involved a trip through a Windows application. I mostly treated the databases that lived on the Psion as clones of the master copy of those on the home network (based on Solaris), so I didn't worry about true synchronization of the information. However, I wanted to do better.

Well, as you will have guessed from

the title of the article, I bought a Palm Vx. I was encouraged by the numbers of these little machines I had seen at a recent USENIX conference. They seemed to be the *gadget du jour*, and because Solaris 8 was installed with a PDA synchronization application, it looked like a hopeful way of ensuring data could move between the Palm and the home system without passing first through Windows.

I am delighted with the Palm. I reckon I was fluent in its handwriting system, Graffiti, in 10 minutes. As a person whose handwriting legibility has been a source of trouble all my life, I was a little worried that my lack of good micro-coordination would be a problem, but it isn't.

Palm OS, its operating system, is well-designed, and using it is mostly intuitive. To be sure, there are some rough edges in some applications. For example, the interfaces for some parts of the Date Book application operate in what I consider to be a counter-intuitive way. I suppose you can't have everything. I am not going to talk about the Palm

itself a whole lot more, because the machine has been around and evolving for a number of years, and you've probably got one or know someone that does. If the machine has passed you by, then look on the Palm Inc. Web site, whose address is at the end of the article.

I'm going to look at the integration of the Palm with Solaris, and use that as a hook to briefly look at a couple of Common Desktop Environment (CDE) applications that come "as standard" with your Solaris system.

## Connecting your Palm

Let's start by taking a look at `sdtpdasync`, the Java program that's used to link the Palm to your Solaris machine. The Palm comes with a cradle that is plugged into a serial interface socket on your machine. Power is plugged into the cradle and you have a ready-made place to put your Palm when you are at work. In a nice touch, the power light that shows when the Palm is connected (and charging) also doubles as a handy place to store your stylus for easy access. Tip: Buy another stylus and keep it in the cradle.

A trick at this point (at least for the Ultra 10) is to set the speed on the Palm for the direct serial line to 115,200 baud, since the Ultra 10 seems to cope happily with this line speed, and you won't object to driving the line as fast as it can go.

The Solaris PDA Synchronization application is started from an action in CDE (look in *Application Manager/Desktop Apps* and scroll down looking for `PDASync`). I've moved the icon onto my desktop so it's always handy. When you first launch `PDASync` it asks you for the name you have given your Palm for identification purposes, and then just sits there waiting for you to press the Hot Sync button on the Palm cradle.

Before you do that, you need to tell `PDASync` the name of the serial line to which you've connected your Palm. There's a possible problem here, `PDASync` is run by you as a "normal mortal" with your normal access rights. If the file permissions on the device in the file system tree that maps onto the terminal line do not allow you to have read/write access to the port, then `PDASync` will tell you that the "Port is in use by another application."

If you look at say, `/dev/term/b` with `ls -l`, you'll find that it's a symbolic link to the "real" device created by the system set-up process. On my system it links to `../../../../devices/pci@1f,0/pci@1,1/ibus@1/se@14,400000:b`. To find the real device:

```
$ cd /dev/term
```

then

```
$ cd ../../../../devices/pci@1f,0/pci@1,1/ibus@1
```

Having done this command, you may realize that when you are in `/dev/term`, `../../../../devices` is actually `/devices`. But nevermind, you got there. Now use `ls -l` to look at `se@14,400000:b`. If you don't have read/write access rights on the device, then `PDASync` cannot access it and it won't work. You will probably need to have superuser rights to change its

permissions. I just set mine to enable all the read/write permissions on the device.

My Palm can also use PPP (Point-to-Point Protocol)—or SLIP or CSLIP—over the serial line to reach out to the Internet. I've tried using Solaris PPP and it works. You need to understand how to configure PPP and its associated IP routes to have the link functioning, and sadly, it's not for the faint-hearted. I did find that my SunPCi card was not terribly happy to be sharing the machine with PPP—it started to sulk and refused to talk to the network. It turns out that it's necessary to tell its start-up script the name of the interface that the on-host support daemon should use. This can be done by supplying an environment variable holding the interface name when you start the `sunpci` interface program.

Also, the `PDASync` application doesn't talk to an Internet-connected Palm, so there's a need to be able to run both PPP and the `PDASync` application using the same direct serial link. In fact, the ownership problems with serial devices discussed above materialized when I was testing the use of PPP on the line. When PPP went away, it left a protected terminal port behind. If I was to run PPP and `PDASync` full time, then the trick is to have two special devices in `/dev/term` that refer to the same device. PPP is run and configured to use `/dev/term/b` as before. `PDASync` uses `/dev/term/bb` which is a new device that you will need to create by hand.

The first step to make the new device is to gather the necessary information. If you use `ls -l` on the terminal endpoint in `/devices/pci@1f,0/pci@1,1/ibus@1`, you'll see that the first column of the output will read something like:

```
crw-rw-rw-
```

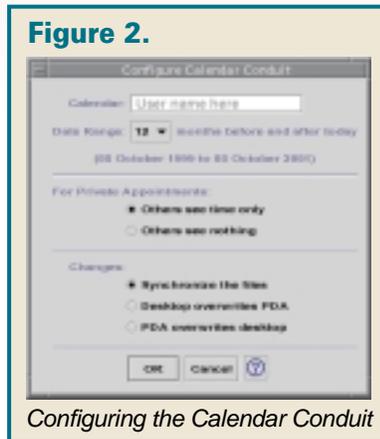
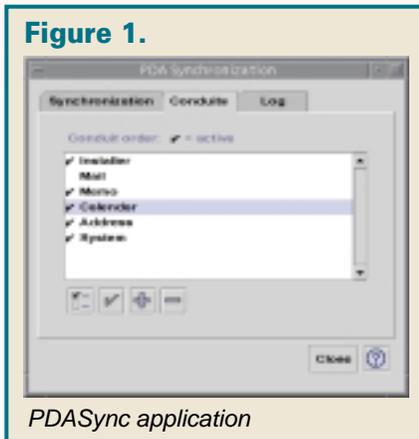
The important thing here is the "c" at the start of the line showing that the file is a "character special device." A little further along the `ls` output, between the group name and the modification date, you'll see a pair of numbers. On my machine, they are "20,1" for `se@14,400000:b`. It's this pair of numbers that the kernel uses to route information to the specific terminal line. We can now clone the device (again these commands need superuser access rights):

```
# cd /dev/term
# /usr/sbin/mknod bb c 20 1
```

creates a new character special device called `bb` that accesses the terminal line. This new device endpoint can be set to be accessible by mortals and used by `PDASync`. Caveat: the PPP system sets the device it uses as non-accessible because it doesn't want coffee-crazed night workers sending garbage down the link or interfering with communication by reading from the link. By establishing a clone device, you're circumventing a safety feature.

## PDASync

The `PDASync` application creates a place for itself on your home directory called, somewhat not surprisingly, `PDASync`. In this directory you'll find another, called `PalmPilot`, which is where all the action happens.



Files are copied between the Palm and the Solaris file system using “conduits.” There are six. Each is separately selectable, so you don’t have to run them all for each hot sync operation. Turning them on and off is done by selecting the conduit and clicking on the box containing a tick (see Figure 1, PDASync application) to toggle a flag. It really would help if you could select them all to be on or all to be off. Each conduit is also configurable, again by selecting the appropriate line in the menu, and clicking on the menu select button to the left of the tick. I think the artwork for this button is somewhat enigmatic. For conduits that can move data in both directions, you usually select that the data will be synchronized, with each end being updated from the other. You can also tell it that one end (or the other) is to be treated as the master. Figure 2, Configuring the Calendar Conduit, shows the configuration options for the *Calendar* Conduit.

As I said, there are six conduits, each with a specific task. *Installer* enables you to download applications to the Palm. They are installed and an icon will appear on your Palm. Files for installation are placed in the `install` directory of the Palm Pilot tree. Once they are moved to the Palm, they are deleted from the Solaris system, so you don’t end up re-installing things. Also, the installation system ignores anything that it doesn’t recognize (it uses the file suffix to work this out). The selectivity is handy because you can store all the master zip files in the `install` directory.

*System* is a backup mechanism. All files are copied from the Palm and placed into the `backup` directory. If your Palm dies or needs a hard reset, you can transfer files from here to the `install` directory, perform a hot sync and be up and running. Transferring files for backup can be slow, since the program doesn’t seem to have any capacity for checking what it needs to copy and then doing the minimum work. It just moves everything, every time. It’s safe, but tedious.

*Memo* moves the files that make up the Palm’s Memo system to corresponding sub-directories in the `memos` directory. A sub-directory structure that maps onto Palm’s categories is created, and the result is a little tree of text files on the Solaris system that you can deal with in whatever way seems sensible to you. You can add files to the Palm by plonking copies of them into the directory. Deleting a file from a directory will cause the corresponding file to be deleted from the Palm during the

hot sync operation.

*Calendar* synchronizes the Solaris Calendar system with the Palm’s DateBook application. It all seems to work cleanly, although it does seem that if you have the Calendar active on the CDE desktop, sometimes the Calendar sync fails to open the Solaris Calendar manager and nothing is done. It pays to watch the operations log in the PDASync log window to see what has happened. I’ve usually found that performing another sync operation moves information the second time around. On one occasion, I found that deleting an entry on the Palm was not reflected back on the main Solaris Calendar, but all my attempts to replicate the problem have failed. So there must have been something funny going on at the time.

*Address* synchronizes Address Card records in the Solaris Address Manager Card system with the Palm’s Address application. Actually, the Palm supports less information than the Solaris system, and there’s a certain amount of shoehorning done to get the data onto the Palm. You’ll find more about this application below.

Finally, *Mail* can move email messages from your Solaris mailbox onto the Palm, and can also take any messages that you compose and send them. The Mail application on the Palm can also directly use TCP/IP if you have configured PPP between the Palm and the Solaris system. I’ve only really played with this application largely because I don’t use Sun’s CDE mail system.

Well, what’s missing? The Palm has a very nice To-do list mechanism where you can create ordered lists of things to do, set priorities and attach notes. The system is brilliant for both shopping lists and also for working out what you need to do next. I’d love to be able to get these lists out of the Palm, perhaps by dumping them to text files that I can process and use on my Solaris system. Because you can have several named lists running at any one time on the Palm, there’s an impedance mismatch between the To-do list mechanism provided by Sun’s Calendar manager. I’ll bet that someone looked at the Calendar system when PDASync was being designed and said “no way.” However, the simple approach of creating a set of text files would be fine for me. I guess my next step will be to attempt to find some UNIX tools that can be used to pull apart the Palm internal database structure, so I can pick at the To-do file created in the backup system. I am sure these tools exist, somewhere.

## Solaris Applications

As you will have realized, I’ve begun to buy into two Solaris applications that have been lying around my Solaris desktop, but have been completely unused for some number of system releases. The first is the Sun’s Calendar system. When you click on the date icon on your desktop, you invoke a graphical front-end that communicates with a daemon that does the job of managing the calendars on the machine. Using a centralized database on each machine makes sense when you are attempting to share calendars. Since I’m all alone on my network, I don’t see the benefit of shared calendars the system offers, so I find it hard

to judge how well the system works in a shared environment.

Perhaps my biggest beef with the Calendar system is that the GUI is boring. It's a standard CDE application, and on my system the calendar is displayed using small white lettering on a grey background. Calendars are an area where the use of color makes sense. You can color code days of the week, and also categorize different types of events using color.

However, in terms of functionality, the system seems fine. It does all the jobs I need it to do. It has significantly more functionality than my previous calendar system. I wrote XCa1 around 10 years ago as an exercise in writing an X application. One problem is that the entries are simply text, and contain no meta information such as times or knowledge of repeated events. I've lived with the restriction for years, but the data doesn't interface well to the current raft of calendar programs that require extra meta information about the strings that they store. It was time for a change, so XCa1 was consigned to the "used and gone" pile. If I ever get the time, it might be interesting to see if it could be used as a frontend to Sun's calendar system.

The second application that I've begun to use is the Address Manager system. The Palm Address application will synchronize with the personal address book data that Solaris stores. The information for every person you enter into the system is stored in `$HOME/.dt/Addresses` with each person's details in a separate file. Sun has at least not invented a new format for these files, which certain other manufacturers would have done. The data is stored in Vcard 2.1 format that was created by the Internet Mail Consortium. This underlying format is flexible, and Sun has used it intelligently.

The main trouble with this type of system is that it rarely copes with the vagaries of human life. For example, the filename used to store the information is "lastname, firstname" format. The filename sounds OK, but I certainly know several distinct individuals who have the same name. At least the system on Solaris permits a home address and a work address, recognizing that most people have at least two contact locations. But it's also true that many people have more than just one work and home phone number, and can have several email addresses, or several URLs. A contact system needs to be a bin in which you can throw any information that you like, and have it appear later; and of course, the requirement of ultimate flexibility is hard to get right.

That being said, Sun's system has a chance of being flexible. When you edit the contact address, the data is placed in a window that contains keywords like: "Work phone" and if text is entered after this keyword it will find its way into the Vcard as

```
TEL;WORK:nnnnnnnnnn
```

when you save the information. If you create a second line that says "Work Phone" and give it a number, the GUI accepts it and doesn't complain. However, it silently just writes the second one to the file, which isn't very useful. I don't see why the system cannot display lines with replicated keywords. It's just a matter of coding.

The address system on the Palm is more limited than that on Solaris. It only permits one postal address per person and this

maps onto the work address from the Solaris system. It's not really a problem for me. I usually just want the set of phone numbers when I am out and about, and the Palm supports both home and work primary phone numbers. When you synchronize Palm address data with Solaris, you'll find that the Vcard contains several fields that contain Palm extensions to the Vcard protocol. The four custom fields and the country field are extensions to the protocol marked as `X-PalmPilot`, for example. Sun has bent the system to be nicer to Palm users.

## Finally

I haven't mentioned anything about the time logging function I had on the Psion and needed to replace. I began by poking around on the Internet and found that there is a considerable amount of software generation surrounding the Palm. Several software houses sell products, but more interestingly, there's a significant freeware and shareware culture. There are several possible time logging systems, many of which do much more than I need.

Sadly, all of the programs that I looked at failed to understand that workaholic computing people regularly work over midnight. I was somewhat surprised to find that the inability to deal with working over midnight affected the commercial products that I tested, and most of these seem to do everything except make the coffee. The "everything" usually demands access to a Windows system and this defeats the object somewhat. Actually, my original Psion application also failed to log events over midnight but the authors finally fixed that after a number of bug reports.

I've ended up with a simple program called Take Time that just logs events. It tags logged times with clients and project names. Take Time exports its database on demand by writing a log file into the Palm's Memo system. I pick the file up on Solaris and process it with Perl. Take Time is by Andrew Duffy and is freeware. I have a kludge on Solaris to make the logging work over midnight and have hopes that the problem will be fixed in Take Time itself.

## Where to Find Things

The main portal for the Palm is, not surprisingly, <http://www.palm.com>. The site will lead you onto many other related sites. While I was wondering about conduits for To-do's, I spent some time trying to find more information about PDASync. There was nothing more than the basic information on the Net, and I found nothing on how it works or how to augment it. I was poking around on <http://www.sun.com> with a search string of "Palm," and I found that you can get a Java VM to run on the Palm, which may interest some people.

Information on Vcard format can be found at <http://www.imc.org> and Take Time can be found at [http://www.pathcom.com/~carolron/PDA\\_soft.htm](http://www.pathcom.com/~carolron/PDA_soft.htm). 

---

*Peter Collinson runs his own UNIX consultancy, dedicated to earning enough money to allow him to pursue his own interests: doing whatever, whenever, wherever ... He writes, teaches, consults and programs using Solaris running on an UltraSPARC/10. Email: [pc@cpq.com](mailto:pc@cpq.com).*