



www.photocase.de

## Synchronizing Palmtops with Desktop Computers

# Hand to Desk

A PDA is standard equipment in today's business world. Owners of handheld computers use them to manage appointments, answer email, and even play the occasional game of solitaire. In this article, you'll learn how to transfer personal data from your Palm to your Linux desktop. **BY HAGEN HÖPFNER**

**A**fter you store your contact data and appointments on your Palm, you may wonder how to go about transferring this data to your desktop. The Palm comes complete with a Windows synchronization tool. But what about Linux? Actually, there are several free programs for transferring Palm data to Linux. In fact, there are so many solutions that we can't cover them all in one article. Instead, I will describe a selection of the best programs: KPilot [1], Gnome-Pilot [2], and J-Pilot [3].

As its name suggests, KPilot is a KDE application. Gnome-Pilot synchronizes the Palm with Gnome-based applications, and J-Pilot is a GTK-based application that models its look and feel on the official Palm Windows software tool.

### Locating the Software

Most current distributions should have binary packages for the programs we will examine. But if you are a do-it-yourself type, you can download the sources for pilot-link from [4] and for J-Pilot from [3]. Gnome-Pilot [2] requires a large number of Gnome developer packages, and this makes building the application quite complex. The same thing can be said of KPilot [1], which is part of the *kdepim* package that also includes KOrganizer [6]. SuSE refers to the Gnome-Pilot package as *gnome-pilot*. There are source code and binary packages for Evolution at [5]. Make sure you use a package that was specially pre-compiled for your distribution.

### Basic Command Line Support

All three programs use the pilot-link toolkit, which is available at [4], and because they all use pilot-link, their features are similar. The main differences between the tools are in their ability to pass synchronized data to other applications, such as KOrganizer [5], or Ximian Evolution [6]. Because pilot-link underlies each of these tools, it makes sense to take a look at some pilot-link background information before we go on to the task of getting the Palm to talk the Linux desktop PC.

pilot-link is a collection of shell scripts that help the Linux PC and the Palm to communicate using the serial [7], infrared [8], or USB [9] ports. As it is the most widely used application of this kind, besides ColdSync [10], most Linux distributions should include it. The fact that the current stable version of ColdSync does not support PalmOS5 devices is another convincing argument in favor of pilot-link. Incidentally, if you are not sure if your PDA is supported, you can check out the list of devices with pilot-link support at [11].

After connecting your Palm to the PC, and making sure that the device file (*/dev/pilot*) exists (see "Connecting your Palm to your PC"), you are now ready to start. It makes sense to create a safe backup copy of your data. You can use the first tool in the pilot-link package to do so. But increase the data transfer rate before you start transferring data, as pilot-link defaults to a safe setting with 9600

baud. As the manpage (*man pilot-link*) explains, you can change the transfer speed by setting the *PILOTRATE* environment variable.

If you use the Bash shell, for example, you can *export PILOTRATE = 115200* to up the speed. If you use the tcshell, you would need to *setenv PILOTRATE 115200* instead. This will increase the data transfer speed between the PC and Palm noticeably. A backup with 2.4 MBytes of data would take almost 45 minutes at 9600 baud, compared to five minutes at 115200 baud. First, create a directory for the backup on your desktop:

```
mkdir ~/.my_palm_backup
```

and then launch the backup by typing:

```
pilot-xfer -b ~/.my_palm_backup/
```

The program will then wait for you to press the sync button on the cradle or launch a Palm application to start the synchronization routine. To restore this backup to your Palm device, you would need to type *pilot-xfer -r ~/.my\_palm\_backup/*. Table 1 has an overview of interesting *pilot-xfer* parameters, or you can type *man pilot-xfer* for a complete list.

**Table 1: Useful *pilot-xfer* Parameters**

Parameter	Meaning
-p devicefile	Specifies the device file to use (default: <i>/dev/pilot</i> )
-b directory	Backs up your Palm device to the specified directory
-u directory	Replaces the existing backup data (in <i>directory</i> ) with new Palm data
-s directory	Similar to -u, but removes any data deleted from the Palm from the PC
-r directory	Restores a backup from the <i>directory</i> to the Palm
-i file	Installs PalmOS files (file extensions <i>prc</i> , <i>pdb</i> , <i>pqa</i> ) on the Palm

## The Clone: J-Pilot

Of the three programs we will be looking at, J-Pilot most closely resembles a Windows application. It is not a Java applet, although the name might suggest otherwise, but a C program that uses Gtk toolkit. The J stands for the developer's first name, or his wife's (Judd or Jacki Montgomery). The main window pops up when you type *jpilot* to launch the program (Figure 1).

As previously mentioned, you will want to set the maximum data transfer rate before synchronizing. To do so, use the configuration tool in *File|Preferences* (see Figure 2). The fields for the data transfer parameters are located in the *Settings* tab. Check out the manual at [12] for a list of settings.

After setting the data transfer speed, click the *Sync* button in the main window to tell the PC to listen for the sync data. Then enable HotSync on your Palm or cradle to start the data transfer. You can create a backup by clicking the corresponding button in the main J-Pilot window.

After completing synchronization, your Palm data should be on the PC. Click one of the icons in the main J-Pilot window to access the data.

## Gnome-Pilot for Gnome

Although J-Pilot supports plugins [14] to add functionality for synchronizing the Palm with Gnome applications, Gnome-Pilot [2] is a better choice. The program's core component, *gpilotd*, is a daemon that sits in the background, waiting for the Palm to start synchronizing. To con-

figure the server daemon, you can enter *gpilotd-control-applet* in a console window, or use the Gnome menu entry. If you do not have a configuration to modify, this entry will launch a druid to create a configuration.

The druid takes you to the preferences dialog, where you can set up the communication parameters, such as the correct device and speed. The next dialog is for the credentials. If you have synchronized your Palm previously, you should already have a username. In this case, you can click *Forward* to tell Gnome-Pilot to sync and discover the username. If not, the second option in this dialog allows you to specify a new username.

The next windows prompt you to specify a directory and view the configuration settings.

To fine tune the configuration, use the dialog shown in Figure 3, which will auto-launch when you complete the druid. This dialog is also displayed if you re-run *gpilotd-control-applet*.

Figure 3 shows the Gnome-Pilot configuration tab that controls how Gnome Pilot is synchronized with external applications. The most interesting entries here are the ones for *EAddress*, *ECalendar*, and *EToDo*, which allow you to synchronize Evolution and Gnome-Pilot.

Gnome-Pilot will run on KDE and does

not necessarily require a complete Gnome environment.

## KPilot for KDE

Both Gnome-Pilot and J-Pilot will run independently of your current window manager or desktop environment. But KDE users looking for a convenient sync tool would do well to choose KPilot, which has all the

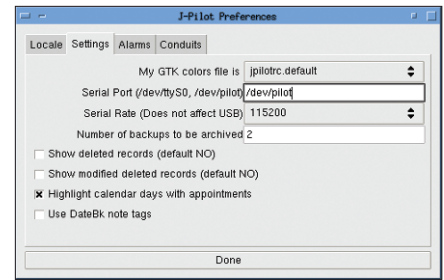


Figure 2: J-Pilot Preferences.

advantages of Gnome-Pilot and J-Pilot in a single tool. Just like Gnome-Pilot, KPilot runs a daemon to listen for Palm synchronization requests, and the GUI

(see Figure 4) has a wide range of features for processing the synchronized data.

Typing *kpilot* or clicking on the menu entry to launch the program, launches the KPilot daemon and opens the main KPilot window at the same time. The window has menu items for displaying synchronized tasks and addresses. It

can also display additional information on the file retrieved from the Palm device in the *Generic DB viewer*.

Before you start, you will need to set up the transfer speed for KPilot in *General setup|Device*. When you launch KPilot for the first time, you may need to add the *Pilot User*, although KPilot can automatically detect the user when you sync for the first time.

## Synchronization Types

You can set up the synchronization type in the *HotSync* tab, specifying if KPilot should create a backup, for example. This is also where you set up a conflict resolution strategy. *Ask User*, the default, is probably your best option.

The *Viewers* tab allows you to configure the behavior of the main KPilot program, specifying if the viewer is allowed to modify entries or simply to display records. In the *Backup* tab you need to specify the databases (Palm files) that KPilot should not synchronize, or not restore in the other direction. The default is to ignore the *Arrng* (Arranger), *PmDB* (HanDBase), *lnch* (application) and *AvGo* (AvantGo) databases. These

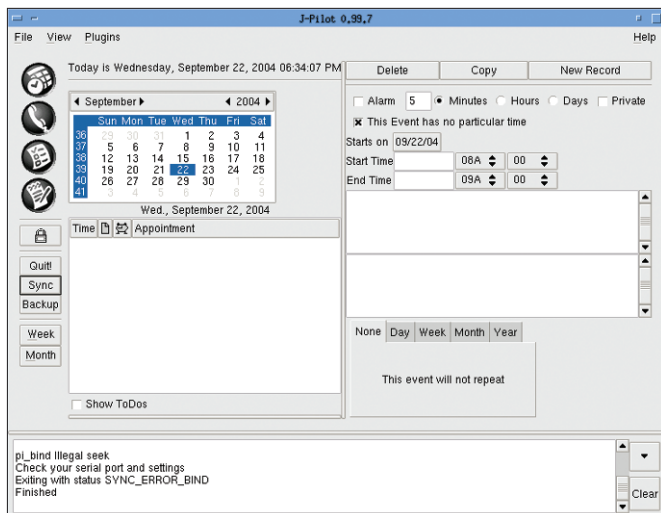


Figure 1: Looks just like the original: J-Pilot.

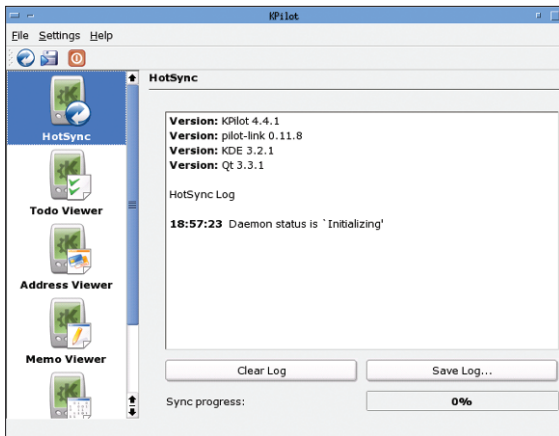


Figure 4: The KPilot GUI.

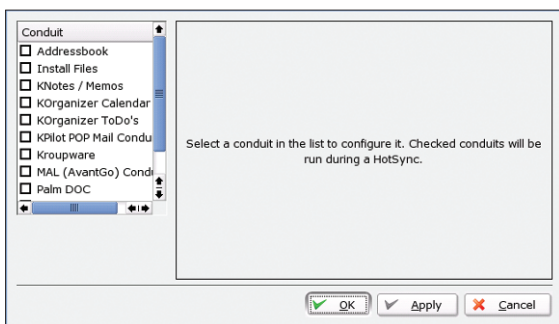


Figure 5: Synchronizing KDE programs with KPilot.

four types are not synchronized, as the databases could be damaged during the process.

You can select more databases to prevent KPilot from synchronizing them. To do so, simply add the database ID to the list by clicking on the “...” button. This pops up a dialog box where you can add

a database, *addr* for example, and then check the checkbox for that database. The next time you synchronize, KPilot will ignore the address database.

After completing the basic configuration, you can sync for the first time. To do so, launch the sync application on the Palm, or press the HotSync button on the cradle as you did with Gnome-Pilot. KPilot shows a sync progress indicator in the main window (see Figure 4).

It is useful to be able to display and modify the synchronized data in the appropriate KDE programs, and that takes us to the *Conduits Setup...* (see Figure 5). There are far too many options for us to go into detail on all of them, and your best approach is to experiment, but I will examine what I consider to be the three most important options.

The most important dataset is probably the one containing your memos (Palm Memos on the handheld), which you will want to synchronize with KNotes. To do so, first enable the appropriate plugin, *KNotes/Memos*, by checking the checkbox. Finished! The

next time you synchronize, your memos will appear in KNotes.

It is just as simple to sync appointments with KOrganizer. Simply enable the plugin and synchronize. Of course, you can configure the plugins to suit your individual requirements.

When you enable a plugin, the right-hand panel in the setup window changes to display another set of options. For example, you can select a different KOrganizer file for the KOrganizer plugin and also specify a conflict resolution strategy. The “ToDo’s KOrganizer” plugin takes the tasks stored on your Palm to your desktop, and the “Addressbook” entry will take care of your address entries. ■

## Connecting Your Palm to Your PC

pilot-link supports a wide range of options for connecting your Palm to your PC, assuming that Linux has driver support to match. The Palm Vx that we tested can use either the serial or infrared port to connect to the PC. The simplest approach is to use the cradle, which attaches to the serial port. All you need then is a symbolic link to the correct device file, and a user account with read and write privileges. Assuming that you will be using the first serial port, you can type `ln -s /dev/ttyS0 /dev/pilot` to create a symlink. The device name for the second com port would be `ttyS1` instead of `ttyS0`.

The privileges you need to assign depend on your distribution. For SuSE, simply run YaST2 to add a user to the `uucp` group. Other distributions may require some manual configuration work. In the worst case, you may need to `chmod +rw /dev/ttyS0` (or `ttyS1`)

to assign all users read and write privileges for the device. You need to be root to do so.

The procedure is similar if you decide to use the infrared port. If this port is not set up, refer to the howto at [8] for details. Then, working as root, run the `/etc/init.d/irda start` script, which should set up the `/dev/ircomm0` device file. Additionally, you can run `findchip` to find out if your PC actually has an IrDA chip with Linux support. If all of these conditions are fulfilled, you can follow the same procedure as for the serial port. Create a symbolic link by typing `ln -s /dev/ircomm0 /dev/pilot`, and assign appropriate permissions.

Newer Palm models have a quicker USB-based connector. In this case, the device file will be `/dev/ttyUSBx`, where `x` reflects the USB port number, for example, `/dev/ttyUSB0` for the first USB port. If the USB refuses to cooperate, try the howto at [9].

## INFO

- [1] KPilot homepage:  
[http://www.slac.com/pilone/kpilot\\_home/](http://www.slac.com/pilone/kpilot_home/)
- [2] Gnome-Pilot project page:  
<http://cvs.gnome.org/viewcvs/gnome-pilot/#dirlist>
- [3] J-Pilot homepage:  
<http://www.jpilot.org/>
- [4] Pilot-Link homepage:  
<http://www.pilot-link.org/>
- [5] Evolution homepage:  
<http://www.ximian.com/products/evolution/>
- [6] KOrganizer homepage:  
<http://korganizer.kde.org/>
- [7] Synchronization via RS232 (serial):  
<http://www.redbug.uklinux.net/palm/pc-connect.html>
- [8] Synchronization via infrared:  
<http://howto.pilot-link.org/irdasync/>
- [9] Synchronization via USB:  
<http://pilot-link.org/README.usb>
- [10] ColdSync, a possible alternative:  
<http://www.coldsync.org/>
- [11] Palm Device Matrix:  
<http://wiki.pilot-link.org/title/Palm+Device+Matrix>
- [12] Henrik Becker: J-Pilot manual:  
<http://www.henrikbecker.de/jpilot/jpilot-manual-en.html>
- [13] Pimlico Software, Inc.:  
<http://www.pimlicosoftware.com/index.htm>
- [14] Synchronizing J-Pilot and GnomeCard:  
[http://spsc.inw.tugraz.at/christian/feld\\_pub/sync.html](http://spsc.inw.tugraz.at/christian/feld_pub/sync.html)
- [15] David H. Silber: Palm OS Desktop HOWTO:  
<http://www.faqs.org/docs/Linux-HOWTO/PalmOS-HOWTO.html>