

Linking Linux to your Bluetooth Mobile Phone

PHONING BLUE

It is becoming increasingly common for new generation mobile phones to have an integrated Bluetooth interface. This article explores how to access your Bluetooth phone using Linux. **BY MARCEL HILZINGER**

Bluetooth has the potential to become one of the most popular toys of the next few years. Because Bluetooth data traffic is transmitted directly from one Bluetooth device to another, Bluetooth does not cost anything. You can easily export wallpaper and ring tones from your PC to your mobile phone, or from your mobile phone to any other Bluetooth device. The Bluetooth protocol can do more than just exchange business cards and photos. It can replace an infrared remote control or even a mouse. Bluetooth is especially interesting for Linux users, as many functions that use USB or the serial port on Windows will only work with Bluetooth under Linux.

This article starts by looking into transferring files between PCs and mobile phones under KDE, and then

explores specific mobile phone models and their features. We used Suse Linux 9.1 and 9.2 for our tests, as well as Fedora Core 3. Suse Linux 9.2 is a good choice for newcomers to Linux because Suse 9.2 YaST has a separate Bluetooth module.

Contact

To get your Bluetooth cellphone talking to Linux, you need a PC with an internal Bluetooth adapter or a so-called Bluetooth dongle for a USB port. Support for USB Bluetooth adapters was introduced with Linux kernel 2.6. If you have kernel 2.4, many adapters will either not work at all or will require a kernel patch. If in doubt, update your distribution, or at least your kernel. After plugging in the adapter, the next step is to start the Bluetooth system (you need to be *root* to do so) by typing `/etc/init.d/bluetooth start`. You should then type `hcitool dev` to check if Linux has detected your Bluetooth adapter correctly. The output should be a line with the `hci0` device name and a hardware address (for example `00:10:C6:29:2E:13`).

Now that the Bluetooth system is running, you can launch the KDE Bluetooth Framework by typing `kbluetoothd`. Then in Konqueror, type the URL `bluetooth:/` to see if the KDE browser finds your mobile phone (see Figure 1). To do so, you need to enable Bluetooth on your phone first, and the device needs to be visible. Check your vendor documentation for more information on the settings for your phone model.

To send a file to your mobile phone, click on the icon (this is labeled K700i in

our example). This tells you the services that the phone supports (Figure 2). In our case, this means clicking on the **Obex Object Push** icon and then selecting *Open with "kbtobexclient"* in the dialog box that appears. This will either launch the KDE Bluetooth Obex Push Client, or you will be prompted to enter a PIN to attach your PC to the mobile phone.

Now We're Talking

Before your PC can send data to the phone, it first has to attach to the phone. This process, known as *pairing*, is based on a shared PIN code. The PIN code required for pairing is stored in `/etc/bluetooth/pin` on Linux. As the file typically has an entry for *BlueZ* by default, you will need to edit the file before pairing. The file should have a single line with a freely selectable number, without the *PIN:* prefix. If you enter the PIN on your mobile phone, the `pin_helper` program specified in `/etc/bluetooth/hcid.conf` will check if the code matches the figure in `/etc/bluetooth/pin`. After successfully completing pairing, the phone will add your PC to the list of known devices.

KDE gives you a front-end for entering the PIN, assuming the KDE Bluetooth

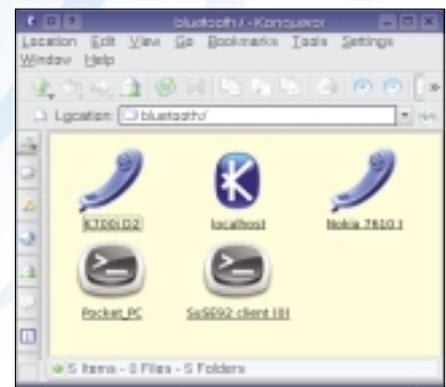


Figure 1: Besides itself (localhost) Konqueror sees two phones, Pocket PC, and another PC.

Framework is installed. Suse Linux 9.1 and 9.2 install the required packages by default. And the KDE Bluetooth Homepage [1] has packages for other distributions.

Change the *pin_helper* program to `/opt/kde3/lib/kde-bluetooth/kblue-pin` for the GUI front-end. The `kdebluetooth` daemon will prompt you for the path when you launch the program. You do not need a static pin with KDE Pinhelper. Just type the PIN in the dialog box (see Figure 3.)

File Transfer

After completing pairing, the KDE Bluetooth Obex Push Client will launch (see Figure 4). You can now select the file(s) you want to transfer in the upper part of the window and drag them onto the *File to send* icon. The device selector should point to your mobile phone. If `Kbtobex-client` fails to find a device, click on *Search* to search for new devices. Clicking on *Send* will transfer your files to the phone. Depending on the phone model, the files will either end up in the phone memory, or you may need to accept the file transfer. The types of files a device will accept depend on its feature set. The following file types worked on all the mobile phones we tested:

- Address book entries (*.vcf)
- Calendar entries (*.ics oder *.vcs)
- Images (*.jpg)
- MIDI ring tones (*.mid)

It is just as easy to transfer files from your phone to your PC. To do so, select the corresponding menu item on the phone; your phone should automatically search for other Bluetooth devices when

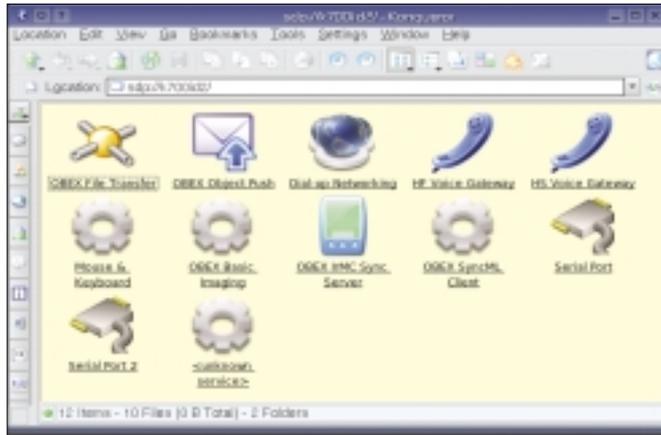


Figure 2: The K700i has a wide range of services. You need to select *Obex Object Push* to send files to the mobile phone.

you do so. Locate your Linux PC by looking for the name you entered in `/etc/bluetooth/hcid.conf` and click on *Send*. Nearly all the devices we tested had a menu item called *Send | Via Bluetooth*. The Motorola V600 and the MDA III were the exceptions. On the Motorola, the menu item can be *Move* or *Copy*, depending on the context. The Windows device does not have a menu item and expects users to select files via the Bluetooth manager (a complicated process that we will be looking at later on.)

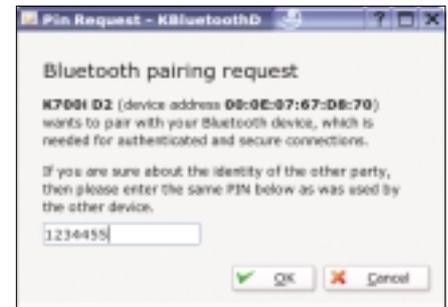


Figure 3: The KDE Bluetooth Framework gives you a GUI-based front-end where you can enter a PIN.

If the KDE Bluetooth daemon is running, a window should pop up on your desktop to indicate that a file transfer is in progress. *Future policy for this device* allows you to opt to automatically accept or reject file transfers in the future. Assuming you accept the file, the window should now show *Incoming File Transfer*. You can either click on the file to open it or select *Save* to save the file.

Users with Fedora Core need the `gnome-bluetooth` and `gnome-vfs-extras` packages to exchange files between their mobile phones and PCs. If Fedora refuses to accept data from the mobile phone, you will probably need to launch

Box 1: Setting Up Bluetooth

The Bluetooth system comprises a number of daemons, although you will typically just need `sdpd` and `hcid`. The *Host Controller Interface Daemon* handles basic Bluetooth functions, and the *Service Discovery Protocol Daemon* tells other communication partners about your device's capabilities. You can launch these daemons via `/etc/init.d/bluetooth start` or manually.

To start the Bluetooth system automatically when you boot your computer, working as `root`, type `insserv bluetooth` on Suse. For Fedora Core, type either `chkconfig --add bluetooth`, or launch the GUI-based editor via *Applications | System settings | Server settings | Services*. Users with Suse Linux will find a graphical *Runlevel Editor* in YaST.

The central configuration file for the Bluetooth system is called `/etc/bluetooth/hcid.conf`. The first thing to do is to change the *name* in *Default settings for HCI devices*, as the default setting for most Linux PCs is *BlueZ*.

For some programs you need to configure the Bluetooth device to emulate a serial port using `rfcomm`. This allows

you to send modem commands to your mobile phone to check the battery or to send a short message. The command for doing this is `rfcomm bind 0 <bt-address>;` replace `<bt-address>` with your Bluetooth hardware address. To allow any user to access the `rfcomm0` device, you need to modify the device privileges. Again working as `root`, type `chmod 666 /dev/rfcomm0` to do so.

As recent Linux systems create device files dynamically, you might need to run this command after every `rfcomm bind` command. If you want to change the access privileges permanently, add the following line to the end of the `/etc/udev/udev.permissions` file (Suse Linux 9.1) or `/etc/udev/permissions.d/50-udev.permissions` (Fedora Core and Suse Linux 9.2):

```
rfcomm*:root::root::666
```

Calling `rfcomm` without parameters displays a list of paired devices. To terminate an existing connection, type `rfcomm release /dev/rfcomm0`. Some mobile phones in our test were unable to send files while they were paired with the `/dev/rfcomm0` interface.

GLOSSARY

Obex Object Push: Obex Push is the alternative name for the Bluetooth Object Push profile. The profile is mainly used for transferring address book and calendar information, but it can handle other file types.

gnome-obex-server first. If you have installed Gnome Bluetooth, again a window pops up asking you to accept the transfer. Fedora stores incoming files in your home directory.

Watch out for the pitfalls with Suse Linux 9.2. If the device is paired, you are not shown a prompt, and any incoming files are automatically stored in `/var/lib/bluetooth`. What this means is that anyone who knows your Bluetooth PIN code can send you files without you noticing.

Blue Competition

In our Linux lab, we tested no less than eight Bluetooth phones (see Table 1). These phones were kindly placed at our disposal by telecommunications providers. There are three groups of phones. The first group has a fairly small display and a proprietary

operating system. The Sony Ericsson T610 and K700i, the Motorola V600, and the Sharp GX15 are all in this group.

The second group is comprised of mobile phones with the Symbian operating system and the Series60 GUI, includ-

ing the Nokia 6600 and 7610 and the Siemens SX1. The third group is a T-Online device, the MDA III with the Windows CE operating system.

With the notable exception of the Nokia 7610, which has a megapixel camera (1152x864), all of these phones have a VGA camera (640x480) that can take still images and 3GP formatted videos. 3GP is a Quicktime format which you can play back on Linux using the Realplayer. If you do not have the Realplayer, just change the file extension from `*.3gp` to `*.mov`. This should allow other players to run the videos. This said, we lost sound output on our computer when we tried this. The Motorola V600 does not have a Video recording function.

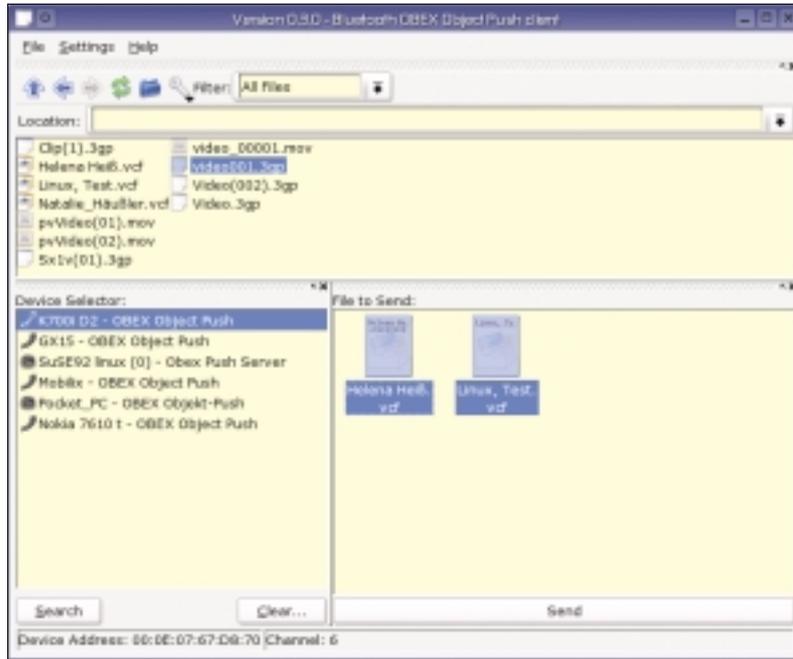


Figure 4: The KDE Bluetooth Obex Push Client allows you to transfer address book and calendar entries to your mobile phone.

Table 1: Overview of Bluetooth Mobile Phones

								
Vendor	Sony Ericsson	Sony Ericsson	Sharp	Motorola	Siemens	Nokia	Nokia	HDC
Model	T610	k700i	GX15	V600	SX1	6600	7610	MDA III
Operating system	proprietary	proprietary	proprietary	proprietary	Symbian - Series60	Symbian - Series60	Symbian - Series60	Windows CE
Internal Memory	2 MB	32 MB	2 MB	5 MB	24 MB	6 MB	8 MB	128 MB
Obex file transfer	✓	✓	-	✓	✓	✓	✓	✓
Bluemote Remote Control	✓	✓	-	-	-	-	-	-
Bemused Remote Control	-	-	-	-	✓	✓	✓	-
IrMC Synchronization	✓	✓	-	-	-	-	-	-
SyncML Synchronization	-	✓	-	-	✓	✓	✓	-
Kmobiletools ¹	✓	✓	-	-	✓	✓	✓	-
Ogg Player	-	-	-	-	✓	✓	✓	✓
SSH Client	-	-	-	-	✓	✓	✓	✓

¹ ✓ means that at least the battery status display worked. Some devices supported SMS and reading of telephone book entries.

ADVERTISEMENT

IrMC Synchronization

The phones that impressed us most in the first group were the Sony Ericsson T610 and K700i. Both phones are simple to use and provide Linux users with the convenient features that Windows users have come to expect. These two phones were the only ones in our lab to support the **IrMC** synchronization protocol, which syncs the data between the phone and the Kaddressbook and Korganizer PIM applications. The Multisync program supports syncing with the GNOME Evolution mail client, although there is no plug-in for version 2.0 of the mail client at this time of writing. This is why we will be focusing on KDE address book synchronization using Kitchensync later on in this article.

As setting up Kitchensync can be a complex process, we will not be looking into building the program in this article. To allow KDE Bluetooth and Kitchensync to interoperate, make sure the KDE Bluetooth Framework was built with the `Irmcsynckonnector` option. Suse Linux 9.2 has ready-to-run RPM packages with this support. In fact, Kitchensync worked fine out of the Suse Linux 9.2 box in our lab. But whatever you do, make sure you create a safe copy of your phone and KDE address books before you start experimenting.

To launch Kitchensync, press `[Alt]+ [F2]` and type `kitchensync`. Now click on *Settings | Configure KitchenSync* and select the *konnector* entry below *Resources*. Click *Add...*, and add the following connectors:

- IrmcSync connector
- Local connector

For the IrmcSync resource, select your phone in *Bluetooth Device*. In the lower left part of the window, specify whether you want to sync your address book or calendar. Each connector will support just one direction (see Figure 5a). To prevent Kitchensync from overwriting your mobile phone book, check the *Read only* option. If your mobile phone is not on the list, it probably does not support IrMC synchronization (see Table 1), or it

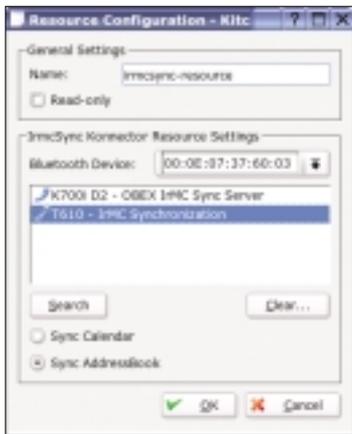


Figure 5a: You mobile phone should be listed as an Irmcsync resource.



Figure 5b: Setting up the KDE address book as the local connector.

may not be set up to be visible.

As the local connector (see Figure 5b), you need to specify an address book and a calendar file. The easiest way of doing this is to click *Select From Existing Resources*. This tells Kitchensync to automatically select your KDE address book and the Korganizer calendar. Again, you can check *Read-only* to prevent Kitchensync from storing entries in your local address book. After adding the connectors you need, click *Edit...* to set up the parameters.

Now select *Settings | Configure Profiles* to specify the profiles that Kitchensync should run. Suse Linux 9.2 gives you two profiles by default: *Syncing* and *Restore Backup*. If you do not have any profiles, select *Add...*, type a name for the profile, and then add the following *Parts*:

- Overview Part
- Connector Backup
- Synchronizer Part

You can experiment with other parts to suit your own requirements. The profiles appear in the main Kitchensync window in the drop down below the menu bar. If you now select the *Syncing* profile, the icons on the left of the window tell you that there are three parts: *Overview*, *Connector Backup*, and *Synchronizer*. The *Restore Backup* profile just has one part called *Connector Backup*. You need *Syncing* for synchronization. Click the icon next to the drop-down (see Figure 6).

GLOSSARY

IrMC: Short for Infrared Mobile Communications. Bluetooth uses a file transfer specification that was originally implemented for infrared connections.

Depending on the size of the phone book, it can take a few minutes for Kitchensync to complete the first sync. The next time you synchronize, the process should be a lot quicker.

If the program detects an entry that looks similar in both address books, it pops up a window like the one in Figure 7. This allows you to specify which entry Kitchensync should use. Unfortunately, Suse Linux 9.2 does not

show a label for the mobile phone button – although the button itself works fine. If you experience difficulty syncing the devices, you might like to add a debugger part to your profile. This allows you to step through the synchronization process and monitor the details.

Remote Control

We were fascinated by the Bluemote [2] remote control software that allows you to remotely control Xmms and OpenOffice presentations using your mobile phone. Before you start experimenting with the remote control, you will need to set up a serial connection to your phone using `rfcomm bind 0 < bt-address>` (see Box 1). The `hcitool scan` gives you the hardware address:

```
kim:~ # hcitool scan
Scanning ...
    00:0E:07:37:60:03    T610
```

Now launch `bluemote` and check if the *Bluemote* heading appears in your phone display. Select *Connections | Tools | PC Remote* to view the individual menu items for the remote control. The `aumix-minimal` program is required to adjust the player volume using the *Volume* control.

Of all the devices we tested, Bluemote only worked with the Sony Ericsson phones. There are no comparable Linux tools for the Sharp GX 15 and the Motorola V600; and these phones generally have a limited range of features.

The Sixties...

The second group of test candidates is the group with the Nokia 6600, Nokia

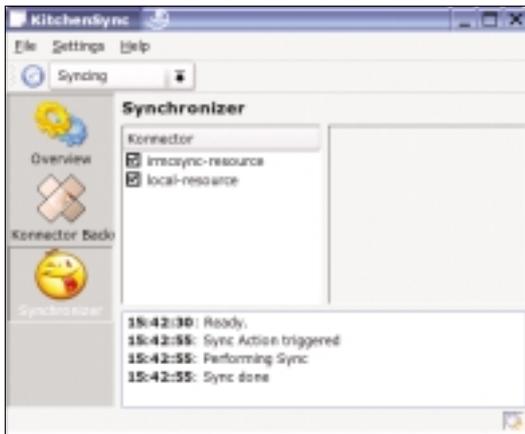


Figure 6: Setting up the KDE address book via the local connector.

7610, and the Siemens SX1 with the Series60 GUI by Symbian. There is no end of free software for Symbian on the Internet, and this means that these three phones work really well with Linux. This said, there does not seem to be a reliable sync program at this time of writing, as these phones only support the newer SyncML protocol. Multisync [3] is a Linux SyncML client, but it only supports syncing with an HTTP server and not direct Bluetooth syncing. For the more adventurous, there is a howto at [4] that describes how to set up an IP/Bluetooth connection with Symbian based phones. This should allow you to sync using Multisync. On the brighter side, work is in progress on a SyncML connector for Kitchensync [5], although we did not have time to test the connector.

There are about 15,000 applications for the Series60 mobile phone, including an Ogg player [6], an SSH client [7], and an NFS server and client tools [8]. The Ogg player and the SSH client run as is, after transferring the SIS files to the phone. Accessing the Series60 phone via the NFS client involves a few manual steps.

Install the `p3nfs-5.16-1.i386.rpm` and transfer the SIS file from `/usr/share/doc/p3nfs-5.16/` to your phone. Then create a `rftcomm` connection by entering `rftcomm bind 0 < bt-adresse> 11` (for details see Box 1). Now launch the NFS client on the phone and then the server by typing `p3nfsd -series60 -tty /dev/rftcomm0`. The NFS daemon will try to mount your phone below `/mnt/psion`. If you prefer to use a different mountpoint (for example, because `/mnt/psion` does not exist on

your computer), stipulate the `-dir /your/directory` when entering the command. The directory should show you the drives on your phone and the memory card as `A:`, `C:`, `E:`. To stop the server, type the following command (ensuring that you specify the correct mount-point):

```
ls /mnt/psion/exit
```

...and their Toys

Bemused [9] brings similar functionality to Symbian phones that Bluemote brought to the

Sony Ericsson models. This is a program for remotely controlling Xmmms or Noatun. Bemused concentrates on navigating the media players; it does not provide a remote control feature for the mouse. The program searches your PC directories for audio or video files, and then goes on to play selected files. You can also download tracks to your phone memory or a memory card, assuming you have enough memory space to do so.

Box 2 gives you an overview of more tools for use with the Series60 and other phones. It may also be worthwhile checking the CD supplied with your phone. Vendor CDs often have games or demo programs that you can install via Bluetooth.

Obex Direct Access

The Series60 phone supports direct access to the file system without the NFS

server workaround. Simply type the `bluetooth:/` address in the Konqueror URL box, select your phone, and then click on *OBEX File Transfer* (see Figure 2).

Depending on the device, you can then drag and drop files between your PC and the phone (and back again) in Konqueror. In contrast to the P3nfs server, this supports parallel transfer of multiple files. Obex file transfer is supported by all the other phones, except the Sharp GX15, although we had some minor difficulties with the K700i and V600. When we attempted to contact the K700i, Konqueror said *Authentication required*. After clicking to clear the message, we were able to move files to the phone but were unable to browse. This is due to an error in Kdebluetooth, which the latest CVS versions have already fixed. With the fix in place you see folders called *Others*, *Images*, *Displayprofile*, *Sounds*, and *Videos*.

The V600 phone displays folders called *audio*, *picture*, and *video* when you click on *OBEX File Transfer*; and a second click opens a folder. However, if you want to copy a MIDI file to the phone to use it as a ring tone, you do not change the directory to *audio*. Instead you just drag the file into the window with the three directories. The phone automatically places the files in the right folder based on the file extension.

The Mega-Phone

The most expensive phone in this test runs Windows CE. With its 128 MBytes

Box 2: Useful Bluetooth Tools

Kmobiletools: Query the battery status and signal strength, read phone book, read and send SMS. Reading and sending SMS is only supported by a few devices. Mobile phone pairing via `rftcomm`. Homepage: <http://kmobiletools.berlios.de>

Gnome Phonemanager: Small tool for sending and receiving SMS. In case of an incoming SMS, a window with the message pops up on the desktop. Only works with Suse Linux 9.2, and with all devices in the test except the Motorola V600 and MDA III. Homepage: <http://usefulinc.com/software/phonemgr/>

Kbthandsfree: Part of the KDE Bluetooth Framework. Supports hands-free speaking using a Bluetooth headset. To use this, launch `kbthandsfree`, select your

phone in the *Bluetooth Service Selection* window (the phone must be visible for this to work) and check the option *Voice over Handsfree*. Now you can simply click *Accept* to accept an incoming call. To make a call, type the number and click on *Dial*. Use the *Settings | Configure Bluetooth Handsfree* menu item to set up the program to automatically store calls in an Ogg Vorbis file.

Multisync: A useful tool for backing up your mobile phone, or syncing the phone with your KDE address book and calendar or with Evolution. Thanks to plug-in support, Multisync can now use various data sources. For example, you can exchange data with an LDAP or SyncML server.



Figure 7: Kitchensync has noticed different phone numbers and spellings.

of RAM, wireless LAN, and a 240 x 320 pixel display, the MDA III is really more of a handheld PC. When you switch the device on, it displays a Windows wizard, which allows you to adjust the screen. Then another wizard appears to introduce you to the touchscreen functions. After completing this step, Windows installs a few files and sets up the desktop. These procedures take about three minutes. As the internal RAM of the MDA III needs a permanent power supply, you have to repeat these steps each time the power supply is interrupted – after changing the battery for example. Windows just forgets all the settings, including the Bluetooth setup.

Whereas all other mobile phones have a menu item called *Send via Bluetooth*, the Windows CE doesn't. First, you need to launch the Bluetooth control center by selecting *Start | Programs | Bluetooth Manager*. Then select *Connections | New | Transfer files*, and select the files you want to transfer. In our tests, this often failed. Instead, Windows reported that

the selected computer did not support the service. If this happens, you need to check if the *Password key required* option is enabled in *Bluetooth | Services | File transfer*, and if so, disable the option.

Unfortunately, the device does not support AT commands, such as the ones used by Kmobiletools to query the battery status or to send short messages. And there is a restricted assortment of free software for the MDA III. At least there is an Ogg Player at [10] and an SSH client at [11]. As far as we are aware, the MDA III does not have anything like the Bemused or Bluemote remote control tool.

In contrast to this, Obex file transfer works well. The Obex file transfer feature supports read and write access to the shared folders on the phone (this is normally just *Documents*). The SyncCE project at [12] also allows you to access the proprietary Windows synchronization protocol. There are also plug-ins for Multisync and Kitchensync, which should support syncing with Evolution or the KDE address book.

Conclusions

Thanks to the useful IrMC client feature, Linux supports a full range of features for the Sony Ericsson T610 and K700i devices. Mobile phones with Symbian OS are useful for budding administrators. These phones may require some manual attention from the user before

you can sync with them, but on the upside, the NFS client even supports scripting.

In our test, it was hard to say anything good, or bad, about the Motorola V600 or the the Sharp GX 15. Despite its full feature set, the MDA III was unconvincing. The fact that the MDA III lacks a *Send via Bluetooth* function makes using the Windows phone far too complex. And the MDA III's battery life of just nine hours is well below par.

While I was writing this article, I managed to sync my own mobile phone, a slightly ancient Ericsson r520, using Kitchensync and to remotely control Kaffeine using Bluemote. Even if your Bluetooth mobile was not explicitly mentioned by this article, you may still have a use for some of these techniques. ■

THE AUTHOR

Marcel Hilzinger studied history at university. He has been working for Suse Linux's Hungarian office in Budapest since 2001, where he translated the Suse documentation into Hungarian, among other things.



INFO

- [1] KDE Bluetooth Framework: <http://kde-bluetooth.sf.net>
- [2] Bluemote remote control: <http://www.geocities.com/saravkrish/progs/bluemote/>
- [3] Multisync: <http://multisync.sf.net>
- [4] IP over Bluetooth for Series60: <http://www.unix-ag.uni-kl.de/~leonard/linux-n6600-howto.html>
- [5] Kitchensync SyncML connector: <http://www.borowka.net/~maciek/ksyncml/>
- [6] Series60 Ogg-Player: <http://symbianoggplay.sf.net>
- [7] Series60 SSH-Client: <http://s2putty.sf.net/>
- [8] Series60 NFS-Client: <http://www.koeniglich.de/p3nfs.html>
- [9] Bemused: <http://bemused.sf.net>
- [10] Ogg Player for Windows CE: <http://www.sto-helit.de/freeware/pocketpc/mortplayer/en/index.pl>
- [11] Windows CE SSH client: <http://pocketputty.duxy.net>
- [12] SynCE project: <http://synce.sf.net/synce/>

Box 3: The Future of Bluetooth

The future of Bluetooth is 2.0 + EDR. The new specification, which the Bluetooth Special Interest Group is promoting, aims to increase the data transmission speed for Bluetooth devices from 80 Kbps at present to 240 Kbps in the future. This would put Bluetooth in a position to support live broadcasting of multimedia files such as audio or video. In other words, you could use your Bluetooth headset to listen to audio CDs or high-quality MP3 files. There are other applications; for example, some drug-stores support Bluetooth transmission for customers who require prints from digital photos.

The new Bluetooth specification is also designed to support multiple associa-

tions. The current adapters allow a maximum of eight nodes in a Bluetooth network. As Bluetooth is becoming more popular for keyboards, mice, digital cameras, and other PC peripherals, this limit of eight nodes is too restrictive for future applications. The new technology is designed to increase the limit to 256 devices.

Finally, Bluetooth SIG is working on improving the security posture of Bluetooth connections and the adapter range. There are some USB adapters that achieve ranges of up to 100 meters in a best case scenario (line of sight connection). The idea is to make this the standard range for Bluetooth devices in the future.