

Mobile Internet access with GPRS

ANYTIME, ANYWHERE

Unpack the *GPRS_Easy_Connect_253_Install.tar.gz* package in a directory of your choice and enter *su* to become *root* (users with Suse Linux 9.2 or earlier can type *sux* to do this). Now change to the *GPRSEC_Easy_Connect_253_Install* directory and launch the install by entering *sh NSTALL*. Click on *Next*, and confirm the terms of the GNU Public License before going on with the install. The next window displays a list of software components the program has discovered. If some of these components are tagged *Error*, you will need to install the missing programs. If you are looking for more details on dependencies, give the *sh INSTALL -c* command when launching the installer.

After fulfilling the dependencies, click *Next* and the installer will copy the required components to their final destinations; you can then select *Finish* to complete the install.

Setup

When you launch the program for the first time, by pressing [Alt] + [F2] and typing *gprsec* in the launcher, the main Easy Connect window appears (see Figure 1.) Click *Setup GPRS Easy Connect* to set up the mobile phone, provider, and connection type (Figure 2.)

Select your phone in the list at the top, and select your provider in *Choose your provider*. Easy Connect displays a picture of the selected phone; this is a big help if you are not sure whether you have selected the correct model from the list. When you select a provider, the program automatically fills out your account data, such as your APN, user name, and password. These details are also available on the web at [2] or directly from your provider.

Permanent mobile Internet access might sound like a nightmare for some, but for others it is a dream come true. All you need is Linux and a fairly recent mobile phone. **BY MARCEL HILZINGER**

General Packet Radio Service (GPRS) is a communication standard for mobile phone networks that is designed for efficient data transfer. GPRS, which is typically faster than older, circuit-switched services, is not intended for voice communication but is, instead, tailored for activities such as email and web browsing over the mobile network. One important quality of GPRS is that it has the effect of being “always on.” You do not have to dial a number in order to access GPRS services.

Many mobile phone networks now offer GPR Service to their users. A tool for called GPRS Easy Connect brings GPR Service to the Linux world. With Linux and GPRS Easy Connect, you are just one click away from mobile Internet access. The current version of this Perl

program supports about 300 mobile phones and around 200 providers. You’ll find a detailed list of supported devices and providers on the project homepage at [1].

At present, the developers Péter Simon, Gábor Pintér and Andrea Bravetti are working on version 3.00, which is based on GTK2 and supports UMTS and EDGE in addition to GPRS.

Perl One...

Installing Easy Connect should be quite simple if you are using any of the current Linux distributions. All you need in order to use Easy Connect is Perl and the *perl-Tk*, *perl-libwww-perl*, and *perl-URI* modules. The Suse Linux DVD has these packages. Users with other distributions may need to download one of the three Perl packages off the Internet.



Figure 1: You need to set up the mobile phone before launching Easy Connect.

Use the *Select the port* option to select an interface for Internet access. The setting depends on your connection type. For Bluetooth devices, you will typically need `/dev/rfcomm0`. (See the box titled “Connecting your phone via Bluetooth.”) Select `/dev/ttyACM0` for USB connections. If you use a serial connection between your phone and your PC, enter `/dev/ttyS0`. If you use your phone’s infrared port, select `/dev/ircomm0`. For information more on setting up Infrared connections, refer to the article in this issue titled “Directory Services: Managing your mobile phone with Gammu and Wammu.”

If your interface type is not on the list, check *Expert port scan* and select the correct entry. You can use the *Select*

your language entry if you prefer to change the interface language. Now click on *OK/Save Data* to display the Easy Connect main window again. The window has a *Connect* button that should allow you to connect. Easy Connect dis-

plays your IP address and data transfer rate statistics when the connection is up.

Usage and Tricks

For initial tests, you should launch Easy Connect as *root*. This gives you access to both `/etc/resolv.conf` (nameserver settings) and to the PPP daemon (chat scripts).

In our lab, our phone often reported problems with the Bluetooth connection, although the connection was active. This can be an expensive issue to resolve when you consider the price of mobile Internet services. To avoid unpleasant surprises, check your phone display after clicking *Connect*: most phones let you know that a connection has been established. For Bluetooth connections where your phone is



Figure 2: To set up the connection, Easy Connect needs details of your phone, provider, and interface.

not connected to your PC, you may even need to confirm the connection.

The `/sbin/ifconfig` command gives you a list of active network devices. As GPRS uses a PPP connection, you should see an entry for the `ppp0` device if the connection is up. If you find this entry, you can type `ping www.kde.org` to see if you can contact the KDE project web server. At average GPRS speeds of 5-10 kbps, it can take a couple of seconds for the first response to start coming in. If `ping` responds with an `unknown host` error, your nameserver settings may be false. Try to ping the KDE website using the IP address in this case:

```
ping 80.232.38.131
```

If the IP address works, you need an entry of the form

```
nameserver IP address
```

to add at least one nameserver to the `/etc/resolv.conf` file. Check [2] for details of your provider's nameserver.

Now that you have successfully established a connection as `root`, you can drop your privileges and repeat the experiment using a normal account. The account you use will need execute permissions for `/usr/sbin/pppd`. The system administrator can grant a user the required permissions by entering the following command:

```
chmod +s /usr/sbin/pppd
```

In many cases, temporary problems can be resolved by switching your phone off and back on.

Unpleasant Surprises

Easy Connect makes child's play of setting up an Internet connection via a mobile phone. However, make sure that

INFO

- [1] Project homepage:
<http://easyconnect.linuxuser.hu>
- [2] GPRS access data:
<http://www.taniwaha.org.uk/gprs.htm>

your mobile Internet connection does not burn a hole in your purse.

Most providers charge a monthly basic rate for the use of GPRS or UMTS – no matter whether you just want to test your connection using `ping` or download 1 MB of data. Packages with larger traffic volumes can cost up to 100 dollars per month. GPRS traffic can cost up to 20 dollars per MB of download volume. UMTS is cheap in comparison: expect to pay about 2 dollars per MB for UMTS service, depending on your contract. Many providers expect you to sign a 2-year contract in order to receive special services such as GPRS. ■

Connecting your Phone via Bluetooth

Current distributions should have no trouble handling Bluetooth setup: simply attach the USB dongle and launch the Bluetooth system by entering `/etc/init.d/bluetooth start` (Suse and Fedora) or `/etc/init.d/bluez-utils` (Ubuntu). Suse Linux 9.3 actually enables Bluetooth services automatically as soon as you attach an adapter. Typing `hcitool dev` will tell you if Linux has correctly identified your dongle. The output from this command should show you the connected device and its Bluetooth address:

```
marcel@kim:~> hcitool dev
```

```
Devices:
```

```
hci0    00:10:C6:29:2E:15
```

You can then enter `hcitool scan` to check for your mobile phone. Note that simply enabling Bluetooth on your mobile will not get you very far. The phone has to be in visible mode. The following example shows that the program has detected both the phone (`Z1010`) and a PC (`ubuntu-0`):

```
marcel@kim:~> hcitool scan
```

```
Scanning ...
```

```
00:0E:07:BF:B4:C4    Z1010
00:04:61:81:5C:6B    ubuntu-0
```

The last step is to set up a serial connection with the mobile; you need to be `root` for this. Again, you will need to run `hcitool scan`; note the Bluetooth address for the phone (watch out for the additional

null between `bind` and the address):

```
rfcomm bind 0 00:0E:07:BF:B4:C4
```

If this command returns `Connection refused`, you probably need to pair the phone with your PC. To do so, enter a four (or more) digit PIN code in `/etc/bluetooth/pin`. Make sure the file contains only the PIN code. Then restart the Bluetooth service and launch your mobile phone's Bluetooth device search function. If the phone prompts you for the PIN code, enter the code from `/etc/bluetooth/pin`.

To test the connection, become `root` and enter `minicom -s`; select `/dev/rfcomm0` as the port, and set the speed to 115200 baud. Now type the AT command `ATI` to output your mobile phone name.

The `rfcomm` command shows you the active connections. To close the connection to the PC, again become `root` and enter `rfcomm release 0`.

Access Privileges: To open a connection with `rfcomm`, you must be `root`. Of course, you will want to access the serial Bluetooth port using a normal user account. To do so, first modify the permissions for `/dev/rfcomm0`. For older Linux distributions that do not use dynamic device files, simply type `chmod 666 /dev/rfcomm0` to do this. More recent distributions create the device file on-the-fly when a Bluetooth adapter is connected. In this case, you can either reset

the file permissions every time you attach a Bluetooth device, or change the Udev settings for the device.

To do so, open the `/etc/udev/rules.d/50-udev.rules` file in your editor, and add the following line (Suse Linux 9.3):

```
KERNEL="rfcomm*", NAME="%k" \
GROUP="uucp" MODE="0660" \
OPTIONS="resmgr"
```

For Suse Linux 9.2, open `/etc/udev/permissions.d/udev.permissions` in the editor and add the following line:

```
rfcomm*:root:users:0666
```

Ubuntu already has a line that starts with `rfcomm[0-9]*`. In this case, you can simply set the permissions to `0666`. Users with Fedora Core 3 don't need to make any changes. The `/dev/rfcomm0` automatically belongs to the logged on user in this case.

Automating connections: To remove the need to type the `bind` command manually, you can tell the Bluetooth system to bind automatically. To do so, edit the `/etc/bluetooth/rfcomm.conf` file. Change the `bind no`; entry to `bind yes`; and enter your phone's Bluetooth address in `device 11:22:33:44:55:66`; for example:

```
device 00:0E:07:BF:B4:C4;
```

Remove the hash sign at the start of the line, if any.