

Managing your mobile phone with Gammu and Wammu

DIRECTORY SERVICES

In today's upwardly mobile age, the mobile phone has become an important information source. Cell phones store information on addresses and appointments, and regular exchanges with home computers keep the data in sync.

The command line tool Gammu [1] lets you manage data on a wide selection of mobile phones – no matter whether you use a data cable, an infrared connection, or a Bluetooth connection. Wammu [2] provides an intuitive front-end to take the hard work out of using Gammu.

Installation

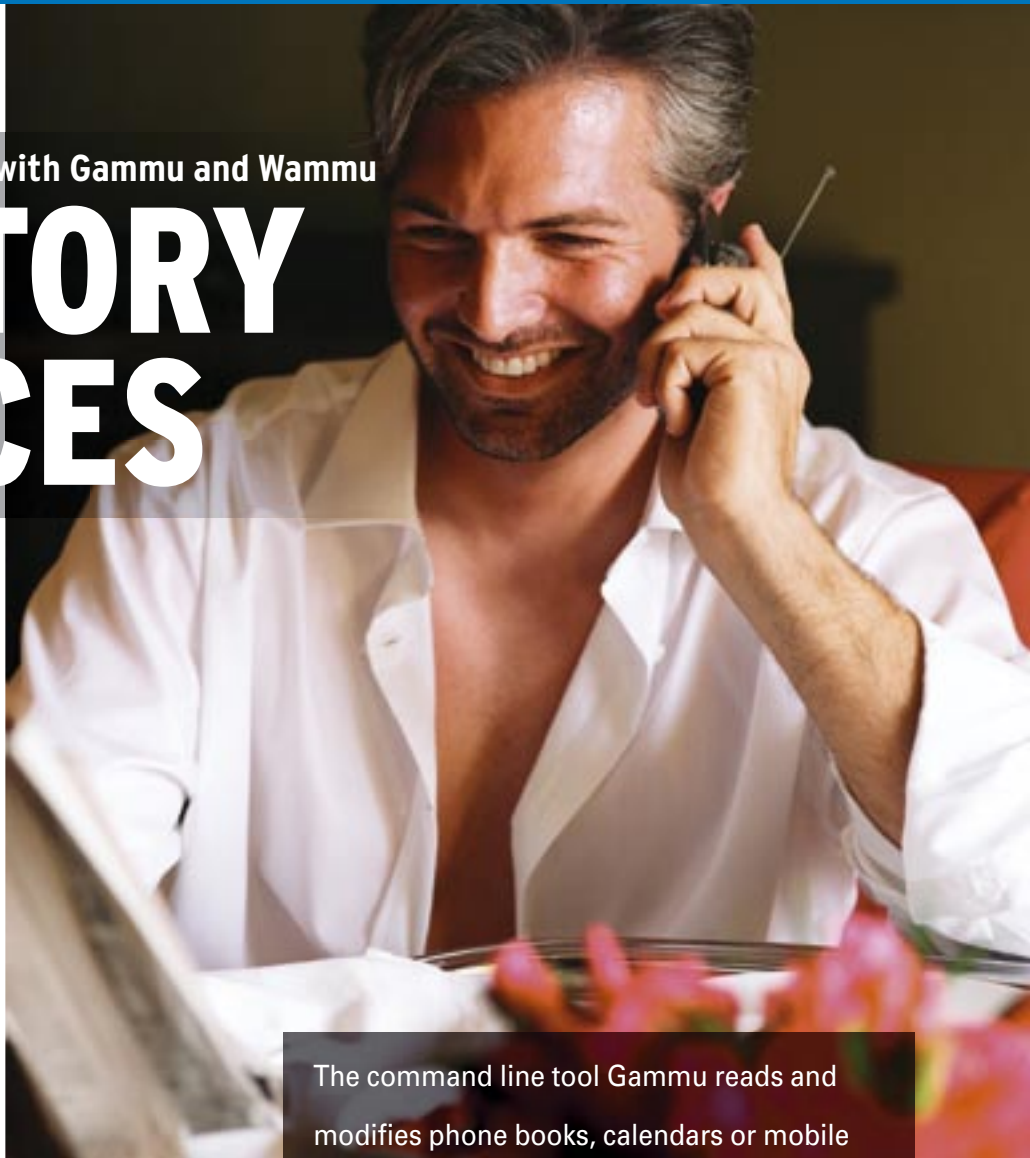
Suse includes both Gammu and the Wammu front-end. Suse users can launch Yast to install both packages. Users with other distributions can check the usual Internet sources, where you'll find the source code, as well as RPM and Debian packages.

Start by downloading the Gammu package from the homepage, and then enter `rpm -Uvh gammu-1.01.11-1.te.i386.rpm` or `dpkg -i libgammu0_1.01.11-0_i386.deb` `gammu_1.01.11-0_i386` for Debian. You will need either the *bluez-libs* (RPM), or *libbluetooth1* (Debian) to provide Bluetooth support for the two packages.

Wammu requires *python-gammu* [3]. Run `tar -xjf python-gammu-0.8.tar.bz2` to unpack the file on your hard disk, and change directory to *python-gammu-0.8*. Then give the following commands to install the package: `python setup.py build` and `python setup.py install`. After completing this step, the steps for installing Wammu are similar.

Wammu

You can now enter *wammu* to launch the GUI. The first time you launch, you



The command line tool Gammu reads and modifies phone books, calendars or mobile phone settings. The GUI-based Wammu tool gives users a convenient front-end for some of the most critical functions.

BY CARSTEN SCHNOBER

will not have a personal *Wammu* configuration file in your home directory; this prompts the program to display the setup dialog. If you have a `~/gammurc` file, Wammu will take the defaults from that file; if not, it will ask you if you would like to autodetect.

If Wammu is unable to autodetect your phone, you will need to enter the data manually in *Wammu* | *Settings* (Figure 1). Type the interface name in the *Device* box; this is `/dev/ircomm0` for infrared, `/dev/ircomm0` for Bluetooth, and `/dev/ttyS1` if you have a USB data cable. Depending on how your distribution handles the configuration, these entries may be slightly different. See the box titled "Mobile Phone with Infrared Interface" for more on configuring an infrared interface. The article titled "Anytime, Anywhere: Mobile Internet Access with GPRS" (in this issue) has a box on configuring a Bluetooth interface.

Define the protocol in *Connection*. There is a list of phones and connection types supported by Gammu at [4]. Then select *Phone* | *Connect* to tell Wammu to connect with the phone. The status bar at the bottom of the window shows the battery status, internal time, and signal strength for the phone.

The *Retrieve* menu is for downloading data from your phone. Not all phones support all the functions, but again, the phone database at [4] has more information. The *Info* menu item reads phone details such as the serial number, model, and firmware version.

Contacts gives you access to your phone book entries, which can be organized in two groups: entries on the SIM

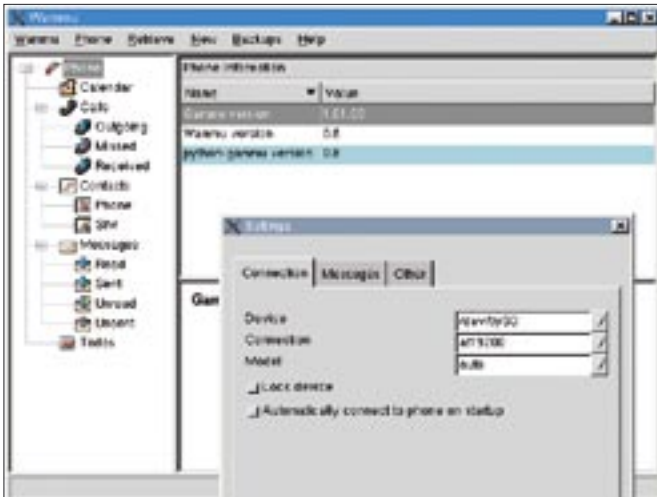


Figure 1: The Wammu user interface gives users access to the contact data and appointments stored on the phone.

card, and entries in the phone's memory. Selecting *Calls* reads incoming and outgoing calls, *Messages* reads SMS messages, *Todos* reads the todo list, and *Calendar* reads the calendar data.

The upload features are not fully supported for all phones in the current Wammu 0.8 version. In our lab, we were unable to create new phone book or calendar entries. New todos also provoked an

error message, but this did not stop them from being stored in the phone's memory. It was a similar story with the SMS editor shown in Figure 2. Despite the error message, the SMS did actually reach its destination.

Safe Copy

A backup of the entries stored on a phone prevents loss of data if the phone is stolen or lost. The *Backup*

Box 1: Mobile Phone with Infrared Interface

Many laptops have an infrared interface, and there are infrared adapters for the USB port. In contrast to a phone-specific data cable, an infrared adapter gives you the advantage of supporting multiple devices. (There are still many phones and computers that do not support Bluetooth, which is a versatile alternative to the infrared option.)

Start by installing the *irda* package; some distributions call this package *irda-utils*. If you need to plug in a USB adapter, don't forget to relaunch the infrared system by entering `/etc/init.d/irda restart`; you need to be root to do this.

Most distributions refer to the infrared interface as `/dev/ttyS1` by default. If you have

a FIR adapter ("Fast Infrared") with Linux support, you will need to change the configuration file entry from `/dev/ttyS1` to the name of your device type, which could be *stir4200* for example. The Suse file that handles these settings is called `/etc/sysconfig/irda`; but names such as `/etc/irda` or `/etc/default/irda-utils` (Debian) are quite common.

The next step is to enter the port you specified for your infrared configuration as the *Port* for your Gammu and Wammu configuration, that is `/dev/ttyS1`, or the name of the device you configured. Add an entry for your phone *Connection*; this is typically *irdaphonet*, although it could be *infrared*, *fbusirda*, or *irdaat*.

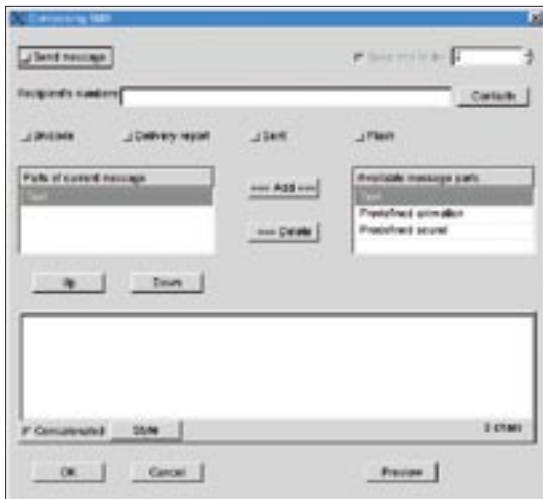


Figure 2: Use the SMS Editor to compose short messages without the mobile phone keypad.

menu helps you save all or a selection of the data downloaded from your phone. Selecting *Backup | Save* opens a file browser where you can specify the export format in the lower selection list.

You can either select *Gammu backup [all data]* to store all data in Gammu's own format, or you can store a subset of the data in the popular *LDIF* or *VCF* standards. More or less any common address book application will support at least one of these. For Nokia devices, you can also use Nokia's proprietary contact data format. Wammu also processes calendar data for use in other programs, writing files either in Apple's *iCal* format or in the *vCal* standard format.

Gammu

Wammu does not support the full range of features provided by a mobile phone, but the Gammu command line tool closes the gaps. Gammu allows users to read the mobile phone's filesystem and control special features, such as internal radios or cameras. You can even run tests to check the vibration alarm and the display or keypad lighting.

Gammu is configured by editing the *.gammurc* configuration file in your home directory, or */etc/gammurc* for all users. If you installed the Gammu RPM, the global configuration file will be located below */etc/*. The Debian package has a compressed sample file below */usr/share/doc/gammu/examples/config/gammurc.gz*. Copy the file to your home directory, run *gunzip gammurc.gz* to unpack, and rename the file by entering *mv gammurc .gammurc*.

To use your phone, you can either become root and edit the system global configuration file, or just edit your personal configuration. The *port* and *connection* entries in the *[gammu]* section are important; a hash (#) tells Gammu to ignore the line. You can enter same data as in the Wammu section earlier, or use data from [4], that is */dev/ttyS1* for *port*, */dev/rfcomm0* for a Bluetooth connection, or */dev/ttyACM0* for a data cable. Also, copy the *connection* entry from Wammu.

The *gammu --help* tool displays a huge list of Gammu options, some of which are device-specific. For more specific information on relevant topics, type something like *gammu --help filesystem*, for example. The *identify* parameter tells you if Gammu supports your phone:

```
$ gammu --identify
Manuf.      : Nokia
Model      : 6610 (NHL-4U)
Firmware   : 4.74 C (14-04-03)
Hardware   : 0640
```

Gammu will give you all the features that Wammu has and more. *gammu --getallcalendar* reads the calendar entries in the phone book and outputs a text. The output you get from the phone book depends on the features your phone supports: some phones will give you the whole phone book when you enter *gammu --getallcategory PHONEBOOK*, and the first entry when you enter *gammu --getcategory PHONEBOOK 1*.

Other phones expect you to specify *--getmemory*, adding a memory type specifier, where *ME* references the device's internal phone book, and *SM* references the SIM card. You can also identify an entry by specifying its number, or use a numeric range. For example, the following would retrieve the address book entries from 1 to 100:

```
gammu --getmemory ME 1 100
```

gammu --getmemory DC 1 - 10 uses a similar technique to access dialed calls stored at memory positions 1 through 10. Specifying *MC* instead of *DC* gives you the missed calls, and *RC* gives you the received calls. *gammu --getmemory ON 1* shows your own number and *VM* lists the messages sent by your mailbox. For all of these functions you will, of

Listing 1: Gammu in Monitor Mode

```
01 $ <b>gammu --monitor<b>
02 Press Ctrl+C to break...
03 Entering monitor mode...
04
05 Enabling info about incoming
   SMS : Enabling info about
   incoming SMS : %s
06
07 Enabling info about incoming
   CB : Enabling info about
   incoming CB : %s\n
08 Enabling info about calls
   : Enabling info about calls
   : %s\n
09 Enabling info about USSD
   : Enabling info about USSD
   : %s\n\n
10
11 SIM phonebook      : Used 017,
   Free 233
12 Dialed numbers    : Used 020,
   Free -08
13 Received numbers  : Used 010,
   Free -254
14 Missed numbers    : Used 266,
   Free -254
15 Own numbers       : Used 000,
   Free 005
16 Phone phonebook   : Used 109,
   Free 191
17 ToDos             : Used 2
18 Calendar          : Used 47
19 Battery status    : 100
   percent
20 Network level     : 100 percent
21 SIM SMS Status    : 8 used, 0
   unread, 40 locations
22 Phone SMS status  : 15 used, 0
   unread, 150 locations, 20
   templates
23 Network reception: Home
   network
24 Network : 262 07 (Network : %s
   (%s", Germany), LAC C75F, CID
   7563
25 Name in phone     : "o2 - de"
```

course, need a phone that supports the memory type you are trying to access.

You can search your phone's filesystem by entering `gammu --getfilesystem`; this gives you all the files stored on the phone sorted by directory. You can then transfer a file to your hard disk by specifying the line number for that file:

```
gammu --getfiles 63
```

`gammu --help nokia` and `gammu --help siemens`, for example, will give you more details on phones by these two vendors. Again, not every phone will support every single command. For example, when we tried to read the security code, by entering `gammu -nokiasecuritycode`, the result was `12345` on several occasions.

If you are interested in uploading or downloading ring tones to or from your phone, you might like to check out `gammu --help ringtone` for the appropriate Gammu parameters. Typing `gammu --getringtoneslist` gives you a complete list of stored ring tones. Typing `gammu`

`--getringtone 51 ringtone`, for example, would transfer the ring tone stored at memory position 51 to a file called `ringtone` on your hard disk. The format depends on your phone type.

Monitoring Incoming Calls

Gammu gives you the ability to monitor incoming calls and SMS messages. The `gammu -monitor` command outputs a list of events at regular intervals, as shown in Listing 1.

In case of incoming short messages or calls, the `monitor` output displays the caller's number. You could write a Bash script to read the data, convert it into a logfile, and evaluate the results with just a little knowledge of programming. Wammu developers, or developers of a new front-end, might use the script in the future to add GUI capabilities.

Depending on the kind of phone (and accessories) you have, Gammu has a wide range of additional features. The `nokia --nokiamakecamerashoot` command triggers the digital camera in your Nokia phone, assuming the phone has

an integrated camera, and `gammu --nokiaturneradio` switches on the radio.

The program works best with Nokia phones; this can be attributed to the fact that the project's predecessor, Gnokii, focused on Nokia. At present, the command line tool lacks a convenient front-end for its excellent range of features. Using the tool requires non-trivial technical skills on the part of the user at times. Although Wammu covers the most common features, other features are not adequately supported. This said, it is still worth your while trying out both tools; if you persevere, Gammu can help you access almost anything stored on your phone. ■

INFO

- [1] Gammu: <http://www.gammu.net>
- [2] Wammu: <http://www.cihar.com/gammu/wammu>
- [3] Python Gammu: <http://www.cihar.com/gammu/python>
- [4] Gammu phone database: <http://www.cihar.com/gammu/phonedb>

Advertisement