Gammu Configuration File

Synopsis

On Linux, MacOS X, BSD and other Unix-like systems, the config file is searched in following order:

1. `$XDG_CONFIG_HOME/gammu/config`
2. `~/.config/gammu/config`
3. `~/.gammurc`
4. `/etc/gammurc`

On Microsoft Windows:

1. `$PROFILE\Application Data\gammurc`
2. `.\gammurc`

Description

Gammu requires configuration to be able to properly talk to your phone. *Gammu Utility* reads configuration from a config file. It's location is determined on runtime, see above for search paths.

You can use *gammu-config* or *gammu-detect* to generate configuration file or start from *Fully documented example*.

For hints about configuring your phone, you can check Gammu Phone Database <http://wammu.eu/phones/> to see what user users experienced.

This file use ini file syntax, see *INI file format*.

Configuration file for gammu can contain several sections - `[gammu]`, `[gammu1]`, `[gammuN]`, ... Each section configures one connection setup and in default mode gammu tries all of them in numerical order. You can also specify which configuration section to use by giving it's number (`[gammu]` has number 0) as a parameter to *Gammu Utility* and it will then use only this section.

`[gammu]`

This section is read by default unless you specify other on command line.

Device connection parameters

`Connection`

Protocol which will be used to talk to your phone.

For Nokia cables you want to use one of following:

`fbus`

serial FBUS connection

`dlr3`

DLR-3 and compatible cables

`dku2`

DKU-2 and compatible cables

`dku5`

DKU-5 and compatible cables

`mbus`

serial MBUS connection

If you use some non original cable, you might need to append `-nodtr` (eg. for ARK3116 based cables) or `-nopower`, but Gammu should be able to detect this automatically.

For non-Nokia phones connected using cable you generally want:

`at`

generic AT commands based connection

You can optionally specify speed of the connection, eg. `at19200`, but it is not needed for modern USB cables.

For IrDA connections use one of following:

`irdaphonet`

Phonet connection for Nokia phones.

`irdaat`

AT commands connection for most of phones (this is not supported on Linux).

`irdaobex`

OBEX (IrMC or file transfer) connection for most of phones.

`irdagnapbus`

GNapplet based connection for Symbian phones, see *Gnapplet Protocol*.

For Bluetooth connection use one of following:

`bluephonet`

Phonet connection for Nokia phones.

`blueat`

AT commands connection for most of phones.

`blueobex`

OBEX (IrMC or file transfer) connection for most of phones.

`bluerfgnapbus`

GNapplet based connection for Symbian phones, see *Gnapplet Protocol*.

`blues60`

Connection to Series60 applet in S60 phones, see *Series60 Remote Protocol*.

New in version 1.29.90.

See also: *Configuring Gammu FAQ*

Device

New in version 1.27.95.

Device node or address of phone. It depends on used connection.

For **cables** or emulated serial ports, you enter device name (for example `/dev/ttyS0`, `/dev/ttyACM0`, `/dev/ircomm0`, `/dev/rfcomm0` on Linux, `/dev/cuad0` on FreeBSD or `COM1:` on Windows). The special exception are DKU-2 and DKU-5 cables on Windows, where the device is automatically detected from driver information and this parameters is ignored.

For **USB** connections (currently only fbususb and dku2 on Linux), you can specify to which USB device Gammu should connect. You can either provide vendor/product IDs or device address on USB:

```
Device = 0x1234:0x5678  # Match device by vendor and product id
Device = 0x1234:-1      # Match device by vendor id
Device = 1.10           # Match device by usb bus and device address
Device = 10             # Match device by usb device address
Device = serial:123456  # Match device by serial string
```

Note: On Linux systems, you might lack permissions for some device nodes. You might need to be member of some group (eg. `plugdev` or `dialout`) or or add special udev rules to enable you access these devices as non-root.
For Nokia phones you can put follofing file (also available in sources as `contrib/udev/45-nokiadku2.rules`) as `/etc/udev/rules.d/45-nokiadku2.rules`:

```
# udev rules file for nokia dku2 devices (udev >= 0.98)
# Put it to /etc/udev/rules.d/ directory
# Taken from http://git.savannah.gnu.org/cgit/gnokii.git/tree/Docs/sample/45-nokiadku2.r
#

ACTION!="add", GOTO="nokiadku2_rules_end"
```

```
SUBSYSTEM=="usb", GOTO="nokiadku2_rules"
# This one is for the old kernels
SUBSYSTEM=="usb_device", GOTO="nokiadku2_rules"
GOTO="nokiadku2_rules_end"

LABEL="nokiadku2_rules"
ATTRS{manufacturer}=="Nokia", MODE="0660", GROUP="plugdev"

LABEL="nokiadku2_rules_end"
```

In case your USB device appears as the serial port in the system (eg. `/dev/ttyACM0` on Linux or `COM5:` on Windows), just use same setup as with serial port.

For **Bluetooth** connection you have to enter Bluetooth address of your phone (you can list Bluetooth devices in range on Linux using **hcitool scan** command). Optionally you can also force Gammu to use specified channel by including channel number after slash.

Before using Gammu, your device should be paired with computer or you should have set up automatic pairing.

For **IrDA** connections, this parameters is not used at all.

If IrDA does not work on Linux, you might need to bring up the interface and enable discovery (you need to run these commands as root):

```
ip l s dev irda0 up          # Enables irda0 device
sysctl net.irda.discovery=1  # Enables device discovery on IrDA
```

Note: Native IrDA is not supported on Linux, you need to setup virtual serial port for it (eg. `/dev/ircomm0`) and use it same way as cable. This can be usually achieved by loading modules `ircomm-tty` and `irtty-sir`:

```
modprobe ircomm-tty
modprobe irtty-sir
```

See also: *Configuring Gammu FAQ*

Port

Deprecated since version 1.27.95.

Alias for `Device`, kept for backward compatibility.

Model

Do not use this parameter unless really needed! The only use case for this is when Gammu does not know your phone and misdetects it's features.

The only special case for using model is to force special type of OBEX connection instead of letting Gammu try the best suited for selected operation:

`obexfs`

force using of file browsing service (file system support)

`obexirmc`

force using of IrMC service (contacts, calendar and notes support)

`obexnone`

none service chosen, this has only limited use for sending file (*gammu sendfile* command)

`mobex`

m-obex service for Samsung phones

`Use_Locking`

On Posix systems, you might want to lock serial device when it is being used using UUCP-style lock files. Enabling this option (setting to yes) will make Gammu honor these locks and create it on startup. On most distributions you need additional privileges to use locking (eg. you need to be member of uucp group).

This option has no meaning on Windows.

Connection options

`SynchronizeTime`

If you want to set time from computer to phone during starting connection.

`StartInfo`

This option allow to set, that you want (setting `yes`) to see message on the phone screen or phone should enable light for a moment during starting connection. Phone will not beep during starting connection with this option. This works only with some Nokia phones.

Debugging options

`LogFile`

Path to file where information about communication will be stored.

> **Note:** For most debug levels (excluding `errors`) the log file is overwritten on each execution.

`LogFormat`

Determines what all will be logged to `LogFile`. Possible values are:

`nothing`

no debug level

`text`

transmission dump in text format

`textall`

all possible info in text format

`textalldate`

all possible info in text format, with time stamp

`errors`

errors in text format

`errorsdate`

errors in text format, with time stamp

`binary`

transmission dump in binary format

For debugging use either `textalldate` or `textall`, it contains all needed information to diagnose problems.

`Features`

Custom features for phone. This can be used as override when values coded in `common/gsmphones.c` are bad or missing. Consult `include/gammu-info.h` for possible values (all **`GSM_Feature`** values without leading `F_` prefix). Please report correct values to Gammu authors.

Locales and character set options

`GammuCoding`

Forces using specified codepage (for example `1250` will force CP-1250 or `utf8` for UTF-8). This should not be needed, Gammu detects it according to your locales.

`GammuLoc`

Path to directory with localisation files (the directory should contain `LANG/LC_MESSAGES/gammu.mo`). If gammu is properly installed it should find these files automatically.

Other options

`DataPath`

Additional path where to search for data files. The default path is configured on build time (and defaults to `/usr/share/data/gammu` on Unix systems). Currently it is used only for searching files to upload to phone using *gammu install*.

Examples

There is more complete example available in Gammu documentation, see *Gammu Utility*.

Connection examples

Gammu configuration for Nokia phone using DLR-3 cable:

```
[gammu]
device = /dev/ttyACM0
connection = dlr3
```

Gammu configuration for Sony-Ericsson phone (or any other AT compatible phone) connected using USB cable:

```
[gammu]
device = /dev/ttyACM0
connection = at
```

Gammu configuration for Sony-Ericsson (or any other AT compatible phone) connected using bluetooth:

```
[gammu]
device = B0:0B:00:00:FA:CE
connection = blueat
```

Gammu configuration for phone which needs to manually adjust Bluetooth channel to use channel 42:

```
[gammu]
device = B0:0B:00:00:FA:CE/42
connection = blueat
```

Working with multiple phones

Gammu can be configured for multiple phones (however only one connection is used at one time, you can choose which one to use with *gammu -s* parameter). Configuration for phones on three serial ports would look like following:

```
[gammu]
device = /dev/ttyS0
connection = at

[gammmu1]
device = /dev/ttyS1
connection = at

[gammmu2]
device = /dev/ttyS2
```

```
connection = at
```

Fully documented example

You can find this sample file as `docs/config/gammurc` in Gammu sources.

```
; This is a sample ~/.gammurc file.
; In Unix/Linux  copy it into your home directory and name it .gammurc
;                or into /etc and name it gammurc
; In Win32       copy it into directory with Gammu.exe and name gammurc
; More about parameters later
; Anything behind ; or # is comment.
; ------------------------------------------------------------------------------

[gammu]

device = com8:
connection = irdaphonet
; Do not use model configuration unless you really need it
;model = 6110
;synchronizetime = yes
;logfile = gammulog
;logformat = textall
;use_locking = yes
;gammuloc = locfile
;startinfo = yes
;gammucoding = utf8
;usephonedb = yes

[gammu1]

device = com8:
;model = 6110
connection = fbusblue
;synchronizetime = yes
;logfile = gammulog
;logformat = textall
;use_locking = yes
;gammuloc = locfile
;startinfo = yes
;gammucoding = utf8

; Step 1. Please find required Connection parameter and look into assigned
; with it device type. With some Connection you must set concrete model

; ================================================================= cables =====
; New Nokia protocol for FBUS/DAU9P
;    Connection "fbus", device type serial
; New Nokia protocol for DLR3/DLR3P
;    Connection "fbusdlr3"/"dlr3", device type serial
; New Nokia protocol for DKU2 (and phone with USB converter on phone mainboard
;                              like 6230)
;    Connection "dku2phonet"/"dku2", device type dku2 on Windows
;    Connection "fbususb" on Linux
; New Nokia protocol for DKU5 (and phone without USB converter on phone
;                              mainboard like 5100)
;    Connection "dku5fbus"/"dku5", device type dku5
; New Nokia protocol for PL2303 USB cable (and phone without USB converter
;                                          on phone mainboard like 5100)
;    Connection "fbuspl2303", device type usb
; Old Nokia protocol for MBUS/DAU9P
;    Connection "mbus", device type serial
; Variants:
```

```
; You can modify a bit behaviour of connection using additional flags
; specified just after connection name like connection-variant.
; If you're using ARK3116 cable (or any other which does not like dtr
; handling), you might need -nodtr variant of connection, eg. dlr3-nodtr.
; If cable you use is not powered over DTR/RTS, try using -nopower variant of
; connection, eg. fbus-nopower.
; -----------------------------------------------------------------------------
; AT commands for DLR3, DKU5 or other AT compatible cable (8 bits, None
; parity, no flow control, 1 stop bit). Used with Nokia, Alcatel, Siemens, etc.
;    Connection "at19200"/"at115200"/.., device type serial
; AT commands for DKU2 cable
;    Connection "dku2at", device type dku2
; ============================================================== infrared =====
; Nokia protocol for infrared with Nokia 6110/6130/6150
;    Connection "fbusirda"/"infrared", device type serial
; Nokia protocol for infrared with other Nokia models
;    Connection "irdaphonet"/"irda", device type irda
; -----------------------------------------------------------------------------
; AT commands for infrared. Used with Nokia, Alcatel, Siemens, etc.
;    Connection "irdaat", device type irda
; -----------------------------------------------------------------------------
; OBEX for infrared
;    Connection "irdaobex", device type irda.
; ============================================================= Bluetooth =====
; Nokia protocol with serial device set in BT stack (WidComm, other) from
; adequate service and Nokia 6210
;    Connection "fbusblue", device type serial
; Nokia protocol with serial device set in BT stack (WidComm, other) from
; adequate service and other Nokia models
;    Connection "phonetblue", device type serial
; -----------------------------------------------------------------------------
; Nokia protocol for Bluetooth stack with Nokia 6210
;    Connection "bluerffbus", device type BT
; Nokia protocol for Bluetooth stack with DCT4 Nokia models, which don't inform
; about services correctly (6310, 6310i with firmware lower than 5.50, 8910,..)
;    Connection "bluerfphonet", device type BT
; Nokia protocol for Bluetooth stack with other DCT4 Nokia models
;    Connection "bluephonet", device type BT
; -----------------------------------------------------------------------------
; AT commands for Bluetooth stack and 6210 / DCT4 Nokia models, which don't
; inform about BT services correctly (6310, 6310i with firmware lower
; than 5.50, 8910,..)
;    Connection "bluerfat", device type BT
; AT commands for Bluetooth stack with other phones (Siemens, other Nokia,etc.)
;    Connection "blueat", device type BT
; -----------------------------------------------------------------------------
; OBEX for Bluetooth stack with DCT4 Nokia models, which don't inform about
; BT services correctly (6310, 6310i with firmware lower than 5.50, 8910,...)
;    Connection "bluerfobex", device type BT
; OBEX for Bluetooth stack with other phones (Siemens, other Nokia, etc.)
;    Connection "blueobex", device type BT.
; -----------------------------------------------------------------------------
;    Connection "bluerfgnapbus", device type BT, model "gnap"
;    Connection "irdagnapbus", device type irda, model "gnap"

; Step2. According to device type from Step1 and used OS set Port parameter

; -----------------------------------------------------------------------------
; Port type | "Port" parameter in Windows/DOS | "Port" parameter in Linux/Unix
; ----------|---------------------------------|--------------------------------
; serial    | "com*:"                         | "/dev/ttyS*"
;           | (example "com1:")               | (example "/dev/ttyS1")
;           |                                 | or "/dev/tts/**" (with DevFS)
;           |                                 | virtual serial ports like
;           |                                 | "/dev/ircomm*" or "/dev/rfcomm*"
; ----------|---------------------------------|--------------------------------
; irda      | ignored (can be empty)           | ignored (can be empty)
; ----------|---------------------------------|--------------------------------
; BT        | Bluetooth device address (example "00:11:22:33:44:55").
;           | Optionally you can also include channel after slash
;           | (example "00:11:22:33:44:55/12"). Can be also empty.
; ----------|---------------------------------|--------------------------------
; dku2      | ignored (can be empty)           | /dev/ttyUSB* or /dev/ttyACM*
; ----------|---------------------------------|--------------------------------
; dku5      | ignored (can be empty)           | connection with it not possible
; ----------|---------------------------------|--------------------------------
; usb       | connection with it not possible | "/dev/ttyUSB*"

; Step3. Set other config parameters

; -----------------------------------------------------------------------------
; Parameter name  | Description
; ----------------|------------------------------------------------------------
; Model           | Should not be used unless you have a good reason to do so.
;                 | If Gammu doesn't recognize your phone model, put it here.
;                 | Example values: "6110", "6150", "6210", "8210"
; SynchronizeTime | if you want to set time from computer to phone during
;                 | starting connection. Do not rather use this option when
;                 | when to reset phone during connection (in some phones need
;                 | to set time again after restart)
; GammuLoc        | name of localisation file
; StartInfo       | this option allow to set, that you want (setting "yes")
;                 | to see message on the phone screen or phone should enable
;                 | light for a moment during starting connection. Phone
;                 | WON'T beep during starting connection with this option.
; GammuCoding     | forces using specified codepage (in win32 - for example
;                 | "1250" will force CP1250) or UTF8 (in Linux - "utf8")
; ----------------|------------------------------------------------------------
; Logfile         | Use, when want to have logfile from communication.
; Logformat       | What debug info and format should be used:
;                 |   "nothing" - no debug level (default)
;                 |   "text"    - transmission dump in text format
;                 |   "textall" - all possible info in text format
;                 |   "errors"  - errors in text format
;                 |   "binary"  - transmission dump in binary format
; ----------------|------------------------------------------------------------
; Features        | Custom features for phone. This can be used as override
;                 | when values coded in common/gsmphones.c are bad or
;                 | missing. Consult include/gammu-info.h for possible values
;                 | (all Feature values without leading F_ prefix).
;                 | Please report correct values to Gammu authors.
; ----------------|------------------------------------------------------------
; Use_Locking     | under Unix/Linux use "yes", if want to lock used device
;                 | to prevent using it by other applications. In win32 ignored

; vim: et ts=4 sw=4 sts=4 tw=78 spell spelllang=en_us
```