sysmocom

sysmocom - s.f.m.c. GmbH

OsmoGGSN User Manual

by Harald Welte

The Asciidoc source code of this manual can be found at http://git.osmocom.org/osmo-gsm-manuals/

HISTORY			
NUMBER	DATE	DESCRIPTION	NAME
1	August 2017	Initial version.	HW

Contents

1 Foreword

Digital cellular networks based on the GSM specification were designed in the late 1980ies and first deployed in the early 1990ies in Europe. Over the last 25 years, hundreds of networks were established globally and billions of subscribers have joined the associated networks.

The technological foundation of GSM was based on multi-vendor interoperable standards, first created by government bodies within CEPT, then handed over to ETSI, and now in the hands of 3GPP. Nevertheless, for the first 17 years of GSM technology, the associated protocol stacks and network elements have only existed in proprietary *black-box* implementations and not as Free Software.

In 2008 Dieter Spaar and I started to experiment with inexpensive end-of-life surplus Siemens GSM BTSs. We learned about the A-bis protocol specifications, reviewed protocol traces and started to implement the BSC-side of the A-bis protocol as something originally called `bs11-abis`. All of this was *just for fun*, in order to learn more and to boldly go where no Free Software developer has gone before. The goal was to learn and to bring Free Software into a domain that despite its ubiquity had not yet seen and Free / Open Source software implementations.

`bs11-abis` quickly turned into `bsc-hack`, then *OpenBSC* and its *OsmoNITB* variant: A minimal implementation of all the required functionality of an entire GSM network, exposing A-bis towards the BTS. The project attracted more interested developers, and surprisingly quickly also commercial interest, contribution and adoption. This allowed adding support for more BTS models.

After having implemented the network-side GSM protocol stack in 2008 and 2009, in 2010 the same group of people set out to create a telephone-side implementation of the GSM protocol stack. This established the creation of the Osmocom umbrella project, under which OpenBSC and the OsmocomBB projects were hosted.

Meanwhile, more interesting telecom standards were discovered and implemented, including TETRA professional mobile radio, DECT cordless telephony, GMR satellite telephony, some SDR hardware, a SIM card protocol tracer and many others.

Increasing commercial interest particularly in the BSS and core network components has lead the way to 3G support in Osmocom, as well as the split of the minimal *OsmoNITB* implementation into separate and fully featured network components: OsmoBSC, OsmoMSC, OsmoHLR, OsmoMGW and OsmoSTP (among others), which allow seamless scaling from a simple "Network In The Box" to a distributed installation for serious load.

It has been a most exciting ride during the last eight-odd years. I would not have wanted to miss it under any circumstances.

— Harald Welte, Osmocom.org and OpenBSC founder, December 2017.

1.1 Acknowledgements

My deep thanks to everyone who has contributed to Osmocom. The list of contributors is too long to mention here, but I'd like to call out the following key individuals and organizations, in no particular order:

- Dieter Spaar for being the most amazing reverse engineer I've met in my career
- Holger Freyther for his many code contributions and for shouldering a lot of the maintenance work, setting up Jenkins - and being crazy enough to co-start sysmocom as a company with me ;)
- Andreas Eversberg for taking care of Layer2 and Layer3 of OsmocomBB, and for his work on OsmoBTS and OsmoPCU
- Sylvain Munaut for always tackling the hardest problems, particularly when it comes closer to the physical layer
- Chaos Computer Club for providing us a chance to run real-world deployments with tens of thousands of subscribers every year
- Bernd Schneider of Netzing AG for funding early ip.access nanoBTS support
- On-Waves ehf for being one of the early adopters of OpenBSC and funding a never ending list of features, fixes and general improvement of pretty much all of our GSM network element implementations
- sysmocom, for hosting and funding a lot of Osmocom development, the annual Osmocom Developer Conference and releasing this manual.

- Jan Luebbe, Stefan Schmidt, Daniel Willmann, Pablo Neira, Nico Golde, Kevin Redon, Ingo Albrecht, Alexander Huemer, Alexander Chemeris, Max Suraev, Tobias Engel, Jacob Erlbeck, Ivan Kluchnikov

May the source be with you!

— Harald Welte, Osmocom.org and OpenBSC founder, January 2016.

2 Preface

First of all, we appreciate your interest in Osmocom software.

Osmocom is a Free and Open Source Software (FOSS) community that develops and maintains a variety of software (and partially also hardware) projects related to mobile communications.

Founded by people with decades of experience in community-driven FOSS projects like the Linux kernel, this community is built on a strong belief in FOSS methodology, open standards and vendor neutrality.

2.1 FOSS lives by contribution!

If you are new to FOSS, please try to understand that this development model is not primarily about “free of cost to the GSM network operator”, but it is about a collaborative, open development model. It is about sharing ideas and code, but also about sharing the effort of software development and maintenance.

If your organization is benefitting from using Osmocom software, please consider ways how you can contribute back to that community. Such contributions can be many-fold, for example

- sharing your experience about using the software on the public mailing lists, helping to establish best practises in using/operating it,
- providing qualified bug reports, work-arounds
- sharing any modifications to the software you may have made, whether bug fixes or new features, even experimental ones
- providing review of patches
- testing new versions of the related software, either in its current “master” branch or even more experimental feature branches
- sharing your part of the maintenance and/or development work, either by donating developer resources or by (partially) funding those people in the community who do.

We’re looking forward to receiving your contributions.

2.2 Osmocom and sysmocom

Some of the founders of the Osmocom project have established *sysmocom - systems for mobile communications GmbH* as a company to provide products and services related to Osmocom.

sysmocom and its staff have contributed by far the largest part of development and maintenance to the Osmocom mobile network infrastructure projects.

As part of this work, sysmocom has also created the manual you are reading.

At sysmocom, we draw a clear line between what is the Osmocom FOSS project, and what is sysmocom as a commercial entity. Under no circumstances does participation in the FOSS projects require any commercial relationship with sysmocom as a company.

2.3 Corrections

We have prepared this manual in the hope that it will guide you through the process of installing, configuring and debugging your deployment of cellular network infrastructure elements using Osmocom software. If you do find errors, typos and/or omissions, or have any suggestions on missing topics, please do take the extra time and let us know.

2.4 Legal disclaimers

2.4.1 Spectrum License

As GSM and UMTS operate in licensed spectrum, please always double-check that you have all required licenses and that you do not transmit on any ARFCN or UARFCN that is not explicitly allocated to you by the applicable regulatory authority in your country.

Warning
Depending on your jurisdiction, operating a radio transmitter without a proper license may be considered a felony under criminal law!

2.4.2 Software License

The software developed by the Osmocom project and described in this manual is Free / Open Source Software (FOSS) and subject to so-called *copyleft* licensing.

Copyleft licensing is a legal instrument to ensure that this software and any modifications, extensions or derivative versions will always be publicly available to anyone, for any purpose, under the same terms as the original program as developed by Osmocom.

This means that you are free to use the software for whatever purpose, make copies and distribute them - just as long as you ensure to always provide/release the *complete and corresponding* source code.

Every Osmocom software includes a file called `COPYING` in its source code repository which explains the details of the license. The majority of programs is released under GNU Affero General Public License, Version 3 (AGPLv3).

If you have any questions about licensing, don't hesitate to contact the Osmocom community. We're more than happy to clarify if your intended use case is compliant with the software licenses.

2.4.3 Trademarks

All trademarks, service marks, trade names, trade dress, product names and logos appearing in this manual are the property of their respective owners. All rights not expressly granted herein are reserved.

For your convenience we have listed below some of the registered trademarks referenced herein. This is not a definitive or complete list of the trademarks used.

Osmocom® and *OpenBSC*® are registered trademarks of Holger Freyther and Harald Welte.

sysmocom® and *sysmoBTS*® are registered trademarks of *sysmocom - systems for mobile communications GmbH*.

ip.access® and *nanoBTS*® are registered trademarks of *ip.access Ltd.*

2.4.4 Liability

The software is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the License text included with the software for more details.

3 Introduction

3.1 Required Skills

Please note that even while the capital expenses of running mobile networks has decreased significantly due to Osmocom software and associated hardware like sysmoBTS, GSM networks are still primarily operated by large GSM operators.

Neither the GSM specification nor the GSM equipment was ever designed for networks to be installed and configured by anyone but professional GSM engineers, specialized in their respective area like radio planning, radio access network, back-haul or core network.

If you do not share an existing background in GSM network architecture, GSM protocols, correctly installing, configuring and optimizing your GSM network will be tough, irrespective whether you use products with Osmocom software or those of traditional telecom suppliers.

GSM knowledge has many different fields, from radio planning through site installation to core network configuration/administration.

The detailed skills required will depend on the type of installation and/or deployment that you are planning, as well as its associated network architecture. A small laboratory deployment for research at a university is something else than a rural network for a given village with a handful of cells, which is again entirely different from an urban network in a dense city.

Some of the useful skills we recommend are:

- general understanding about RF propagation and path loss in order to estimate coverage of your cells and do RF network planning.
- general understanding about GSM network architecture, its network elements and key transactions on the Layer 3 protocol
- general understanding about voice telephony, particularly those of ISDN heritage (Q.931 call control)
- understanding of GNU/Linux system administration and working on the shell
- understanding of TCP/IP networks and network administration, including tcpdump, tshark, wireshark protocol analyzers.
- ability to work with text based configuration files and command-line based interfaces such as the VTY of the Osmocom network elements

3.2 Getting assistance

If you do have a support package / contract with sysmocom (or want to get one), please contact support@sysmocom.de with any issues you may have.

If you don't have a support package / contract, you have the option of using the resources put together by the Osmocom community at http://projects.osmocom.org/, checking out the wiki and the mailing-list for community-based assistance. Please always remember, though: The community has no obligation to help you, and you should address your requests politely to them. The information (and software) provided at osmocom.org is put together by volunteers for free. Treat them like a friend whom you're asking for help, not like a supplier from whom you have bought a service.

4 Overview

4.1 About OsmoGGSN

OsmoGGSN is a Free / Open Source Software implementation of the GPRS GGSN (Gateway GPRS support node) element in side the packet switched core network of 2G and 3G cellular networks.

The GGSN function is the tunnel endpoint on the core network side, from where the external (IP) packet data network

4.2 Software Components

4.2.1 GTP Implementation (libgtp)

The OsmoGGSN source code includes a shared library implementation of the GTP protocol used on the GGSN-SGSN interface. This library and associated header files are installed system-wide and are available to other programs/applications.

In fact, libgtp is what the OsmoSGSN also uses for its use of GTP.

4.2.2 sgsnemu

In order to test OsmoGGSN without running a SGSN and other elements of a cellular network, there is a small command-line utility called **sgsnemu** which is able to simulate the customary operations of a SGSN towards the GGSN, such as a PDP Context Activation.

sgsnemu can even be used for testing against other GGSNs, as the GTP protocol is standardized across implementations.

4.2.3 osmo-ggsn

osmo-ggsn is the actual name of the OsmoGGSN executable program. It implements the GGSN functionality. All parameters are set using the configuration file, by default located in **./osmo-ggsn.cfg**

4.2.4 systemd service file

In **contrib/osmo-ggsn.service** you can find a sample service file for OsmoGGSN which can be used with systemd.

4.2.5 init script

In **contrib/osmo-ggsn.init** you can find a sample init script to be used on systems with classic init process.

4.3 Limitations

OsmoGGSN supports both GTP0 (GSM 09.60) and GTP1 (3GPP 29.060). In the following tables the support of each individual message type is detailed. The numbers before each feature indicates the relevant section in the standard.

4.3.1 GSM 09.60 (GTPv0)

Feature	gtplib	osmo-ggsn	sgsnemu	notes
7.4 Path Management Messages				
7.4.1 Echo Request	Supported	Supported	Supported	
7.4.2 Echo Response	Supported	Supported	Supported	
7.4.3 Version Not Supported	Supported	Supported	Supported	
7.5 Tunnel Management Messages				
7.5.1 Create PDP Context Request	Supported	Supported	Supported	
7.5.2 Create PDP Context Response	Supported	Supported	Supported	
7.5.3 Update PDP Context Request	Supported	Supported	Not	
7.5.4 Update PDP Context Response	Supported	Supported	Not	
7.5.5 Delete PDP Context Request	Supported	Supported	Supported	
7.5.6 Delete PDP Context Response	Supported	Supported	Supported	
7.5.7 Create AA PDP Context Request	Unsupported	Unsupported	Unsupported	
7.5.8 Create AA PDP Response	Unsupported	Unsupported	Unsupported	
7.5.9 Delete AA PDP Context Request	Unsupported	Unsupported	Unsupported	
7.5.10 Delete AA PDP Context Response	Unsupported	Unsupported	Unsupported	

Feature	gtplib	osmo-ggsn	sgsnemu	notes
7.5.11 Error Indication	Supported	Supported	Supported	
7.5.12 PDU Notification Request	Unsupported	Unsupported	Unsupported	
7.5.13 PDU Notification Response	Unsupported	Unsupported	Unsupported	
7.5.14 PDU Notification Reject Request	Unsupported	Unsupported	Unsupported	
7.5.15 PDU Notification Reject Response	Unsupported	Unsupported	Unsupported	
7.6 Location Management Messages				
7.6.1 Send Routeing Information for GPRS Request	Unsupported	Unsupported	Not applicable	
7.6.2 Send Routeing Information for GPRS Response	Unsupported	Unsupported	Not applicable	
7.6.3 Failure Report Request	Unsupported	Unsupported	Not applicable	
7.6.3 Failure Report Response	Unsupported	Unsupported	Not applicable	
7.6.5 Note MS GPRS Present Request	Unsupported	Unsupported	Not applicable	
7.6.6 Note MS GPRS Present Response	Unsupported	Unsupported	Not applicable	
7.5 Mobility Management Messages				
7.5.1 Identification Request	Unsupported	Not applicable	Not applicable	
7.5.2 Identification Response	Unsupported	Not applicable	Not applicable	
7.5.3 SGSN Context Request	Unsupported	Not applicable	Not applicable	
7.5.4 SGSN Context Response	Unsupported	Not applicable	Not applicable	
7.5.5 SGSN Context Acknowledge	Unsupported	Not applicable	Not applicable	

4.3.2 3GPP 29.060 (GTPv1)

Feature	gtplib	osmo-ggsn	sgsnemu	notes
7.2 Path Management Messages				
7.2.1 Echo Request	Supported	Supported	Supported	
7.2.2 Echo Response	Supported	Supported	Supported	
7.2.3 Version Not Supported	Supported	Supported	Supported	
7.2.4 Extension Headers Notification	Supported	Supported	Supported	
7.3 Tunnel Management Messages				
7.3.1 Create PDP Context Request	Supported	Supported	Supported	1
7.3.2 Create PDP Context Response	Supported	Supported	Supported	
7.3.3 Update PDP Context Request	Supported	Supported	Not applicable	1
7.3.4 Update PDP Context Response	Supported	Supported	Not applicable	
7.3.5 Delete PDP Context Request	Supported	Supported	Supported	
7.3.6 Delete PDP Context Response	Supported	Supported	Supported	
7.3.7 Error Indication	Supported	Supported	Supported	
7.3.8 PDU Notification Request	Unsupported	Unsupported	Unsupported	
7.3.9 PDU Notification Response	Unsupported	Unsupported	Unsupported	
7.3.10 PDU Notification Reject Request	Unsupported	Unsupported	Unsupported	
7.3.10 PDU Notification Reject Response	Unsupported	Unsupported	Unsupported	
7.4 Location Management Messages				
7.4.1 Send Routeing Information for GPRS Request	Unsupported	Unsupported	Not applicable	
7.4.2 Send Routeing Information for GPRS Response	Unsupported	Unsupported	Not applicable	
7.4.3 Failure Report Request	Unsupported	Unsupported	Not applicable	
7.4.3 Failure Report Response	Unsupported	Unsupported	Not applicable	
7.4.5 Note MS GPRS Present Request	Unsupported	Unsupported	Not applicable	
7.4.6 Note MS GPRS Present Response	Unsupported	Unsupported	Not applicable	
7.5 Mobility Management Messages				
7.5.1 Identification Request	Unsupported	Not applicable	Not applicable	
7.5.2 Identification Response	Unsupported	Not applicable	Not applicable	
7.5.3 SGSN Context Request	Unsupported	Not applicable	Not applicable	
7.5.4 SGSN Context Response	Unsupported	Not applicable	Not applicable	
7.5.5 SGSN Context Acknowledge	Unsupported	Not applicable	Not applicable	
7.5.6 Forward Relocation Request	Unsupported	Not applicable	Not applicable	
7.5.7 Forward Relocation Response	Unsupported	Not applicable	Not applicable	
7.5.8 Forward Relocation Complete	Unsupported	Not applicable	Not applicable	

Feature	gtplib	osmo-ggsn	sgsnemu	notes
7.5.9 Relocation Cancel Request	Unsupported	Not applicable	Not applicable	
7.5.10 Relocation Cancel Response	Unsupported	Not applicable	Not applicable	
7.5.11 Forward Relocation Complete	Unsupported	Not applicable	Not applicable	
7.5.12 Forward SRNS Context Acknowledge	Unsupported	Not applicable	Not applicable	
7.5.13 Forward SRNS Context	Unsupported	Not applicable	Not applicable	

Notes

1) The "Secondary PDP Context Activation Procedure" is not supported.

5 Running OsmoGGSN

The OsmoGGSN executable (`osmo-ggsn`) offers the following command-line arguments:

5.1 SYNOPSIS

osmo-ggsn [-h|-V] [-D] [-c *CONFIGFILE*]

5.2 OPTIONS

-h, --help
Print a short help message about the supported options

-V, --version
Print the compile-time version number of the OsmoBTS program

-D, --daemonize
Fork the process as a daemon into background.

-c, --config-file *CONFIGFILE*
Specify the file and path name of the configuration file to be used. If none is specified, use `osmo-ggsn.cfg` in the current working directory.

5.3 Routing

Operating the OpenGGSN tun device naturally creates a network setup with multiple interfaces. Consider:

- Typical Linux setups prevent forwarding of packets between separate interfaces by default. To let subscribers reach the internet uplink from the tun device, it may be required to enable IP forwarding.
- Having a locally defined address range assigned to the tun device requires either sensible routing for this address range, or that masquerading is enabled to allow your single uplink IP address to "proxy" for the tun.

These are decisions to be made on a network administration level.

In a trivial case where you have a single box serving GPRS to few subscribers on an arbitrary IP address range not known in the larger network, the easiest way to enable GPRS uplink would be to enable IP forwarding and masquerading.

To manually enable IPv4 forwarding and masquerading ad-hoc, you can do:

```
sh -c "echo 1 > /proc/sys/net/ipv4/ip_forward"
iptables -t nat -A POSTROUTING -o '*' -j MASQUERADE
```

(You may want to replace `*` with the network device name, like `-o eth0`)

There are various ways to enable these settings persistently, please refer to your distribution's documentation — e.g. look for @net.ipv4.ip_forward=1@ in @/etc/sysctl.d/@, and https://wiki.debian.org/iptables for masquerading.

5.4 Multiple instances

Running multiple instances of `osmo-ggsn` is possible if all GGSN instances are binding to different local IP addresse and all other interfaces (VTY, OML) are separated using the appropriate configuration options. The IP based interfaces are binding to local host by default. In order to separate the processes, the user has to bind those services to specific but different IP addresses.

The VTY and the control interface can be bound to IP addresses from the loopback address range.

Example: Binding VTY and control interface to a specific ip-address

```
line vty
 bind 127.0.0.2
ctrl
 bind 127.0.0.2
```

Also make sure to place each instance's GTP bind on a separate IP address (GTP uses a port number that is fixed in the GTP specifications, so it will not be possible to pick differing ports on the same IP address), like:

```
ggsn ggsn0
 gtp bind-ip 127.0.0.2
```

6 The Osmocom VTY Interface

All human interaction with Osmocom software is typically performed via an interactive command-line interface called the *VTY*.

Note

Integration of your programs and scripts should **not** be done via the telnet VTY interface, which is intended for human interaction only: the VTY responses may arbitrarily change in ways obvious to humans, while your scripts' parsing will likely break often. For external software to interact with Osmocom programs (besides using the dedicated protocols), it is strongly recommended to use the Control interface instead of the VTY, and to actively request / implement the Control interface commands as required for your use case.

The interactive telnet VTY is used to

- explore the current status of the system, including its configuration parameters, but also to view run-time state and statistics,
- review the currently active (running) configuration,
- perform interactive changes to the configuration (for those items that do not require a program restart),
- store the current running configuration to the config file,
- enable or disable logging; to the VTY itself or to other targets.

The Virtual Tele Type (VTY) has the concept of *nodes* and *commands*. Each command has a name and arguments. The name may contain a space to group several similar commands into a specific group. The arguments can be a single word, a string, numbers, ranges or a list of options. The available commands depend on the current node. there are various keyboard shortcuts to ease finding commands and the possible argument values.

Configuration file parsing during program start is actually performed the VTY's CONFIG node, which is also available in the telnet VTY. Apart from that, the telnet VTY features various interactive commands to query and instruct a running Osmocom program. A main difference is that during config file parsing, consistent indenting of parent vs. child nodes is required, while the interactive VTY ignores indenting and relies on the *exit* command to return to a parent node.

Note

In the *CONFIG* node, it is not well documented which commands take immediate effect without requiring a program restart. To save your current config with changes you may have made, you may use the `write file` command to **overwrite** your config file with the current configuration, after which you should be able to restart the program with all changes taking effect.

This chapter explains most of the common nodes and commands. A more detailed list is available in various programs' VTY reference manuals, e.g. see [vty-ref-osmomsc].

There are common patterns for the parameters, these include IPv4 addresses, number ranges, a word, a line of text and choice. The following will explain the commonly used syntactical patterns:

Table 1: VTY Parameter Patterns

Pattern	Example	Explanation
A.B.C.D	127.0.0.1	An IPv4 address
TEXT	example01	A single string without any spaces, tabs
.TEXT	Some information	A line of text
(OptionA|OptionB|OptionC)	OptionA	A choice between a list of available options
<0-10>	5	A number from a range

6.1 Accessing the telnet VTY

The VTY of a given Osmocom program is implemented as a telnet server, listening to a specific TCP port. For `osmo-nitb`, this port is `4242`.

Please see Appendix A to check for the default TCP port number of the VTY interface of the specific Osmocom software you would like to connect to.

As telnet is insecure and offers neither strong authentication nor encryption, the VTY by default only binds to localhost (127.0.0.1) and will thus not be reachable by other hosts on the network.

Warning
By default, any user with access to the machine running the Osmocom software will be able to connect to the VTY. We assume that such systems are single-user systems, and anyone with local access to the system also is authorized to access the VTY. If you require stronger security, you may consider using the packet filter of your operating system to restrict access to the Osmocom VTY ports further.

6.2 VTY Nodes

The VTY by default has the following minimal nodes:

VIEW
When connecting to a telnet VTY, you will be on the *VIEW* node. As its name implies, it can only be used to view the system status, but it does not provide commands to alter the system state or configuration. As long as you are in the non-privileged *VIEW* node, your prompt will end in a > character.

ENABLE
The *ENABLE* node is entered by the `enable` command, from the *VIEW* node. Changing into the *ENABLE* node will unlock all kinds of commands that allow you to alter the system state or perform any other change to it. The *ENABLE* node and its children are signified by a # character at the end of your prompt.
You can change back from the *ENABLE* node to the *VIEW* node by using the `disable` command.

CONFIG
The *CONFIG* node is entered by the `configure terminal` command from the *ENABLE* node. The config node is used to change the run-time configuration parameters of the system. The prompt will indicate that you are in the config node by a `(config)#` prompt suffix.
You can always leave the *CONFIG* node or any of its children by using the `end` command.
This node is also automatically entered at the time the configuration file is read. All configuration file lines are processed as if they were entered from the VTY *CONFIG* node at start-up.

Other
Depending on the specific Osmocom program you are running, there will be few or more other nodes, typically below the *CONFIG* node. For example, the OsmoBSC has nodes for each BTS, and within the BTS node one for each TRX, and within the TRX node one for each Timeslot.

6.3 Interactive help

The VTY features an interactive help system, designed to help you to efficiently navigate is commands.

> **Note**
> The VTY is present on most Osmocom GSM/UMTS/GPRS software, thus this chapter is present in all the relevant manuals. The detailed examples below assume you are executing them on the OsmoNITB VTY. They will work in similar fashion on the other VTY interfaces, while the node structure will differ in each program.

6.3.1 The question-mark (?) command

If you type a single `?` at the prompt, the VTY will display possible completions at the exact location of your currently entered command.

If you type `?` at an otherwise empty command (without having entered even only a partial command), you will get a list of the first word of all possible commands available at this node:

Example: Typing ? at start of OsmoNITB prompt

```
OpenBSC> ❶
  show        Show running system information
  list        Print command list
  exit        Exit current mode and down to previous mode
  help        Description of the interactive help system
  enable      Turn on privileged mode command
  terminal    Set terminal line parameters
  who         Display who is on vty
  logging     Configure log message to this terminal
  sms         SMS related commands
  subscriber  Operations on a Subscriber
```

❶ Type `?` here at the prompt, the `?` itself will not be printed.

If you have already entered a partial command, `?` will help you to review possible options of how to continue the command. Let's say you remember that `show` is used to investigate the system status, but you don't remember the exact name of the object. Hitting `?` after typing `show` will help out:

Example: Typing ? after a partial command

```
OpenBSC> show ❶
  version       Displays program version
  online-help   Online help
  history       Display the session command history
  network       Display information about a GSM NETWORK
  bts           Display information about a BTS
  trx           Display information about a TRX
  timeslot      Display information about a TS
  lchan         Display information about a logical channel
  paging        Display information about paging requests of a BTS
  paging-group  Display the paging group
  logging       Show current logging configuration
  alarms        Show current logging configuration
```

```
  stats          Show statistical values
  e1_driver      Display information about available E1 drivers
  e1_line        Display information about a E1 line
  e1_timeslot    Display information about a E1 timeslot
  subscriber     Operations on a Subscriber
  statistics     Display network statistics
  sms-queue      Display SMSqueue statistics
  smpp           SMPP Interface
```

1. Type `?` after the `show` command, the `?` itself will not be printed.

You may pick the `network` object and type `?` again:

Example: Typing ? after show network

```
OpenBSC> show network
  <cr>
```

By presenting `<cr>` as the only option, the VTY tells you that your command is complete without any remaining arguments being available, and that you should hit enter, a.k.a. "carriage return".

6.3.2 TAB completion

The VTY supports tab (tabulator) completion. Simply type any partial command and press `<tab>`, and it will either show you a list of possible expansions, or completes the command if there's only one choice.

Example: Use of <tab> pressed after typing only s as command

```
OpenBSC> s❶
show       sms         subscriber
```

1. Type `<tab>` here.

At this point, you may choose `show`, and then press `<tab>` again:

Example: Use of <tab> pressed after typing show command

```
OpenBSC> show ❶
version    online-help history    network    bts         trx
timeslot   lchan       paging     paging-group logging    alarms
stats      e1_driver   e1_line    e1_timeslot subscriber statistics
sms-queue  smpp
```

1. Type `<tab>` here.

6.3.3 The `list` command

The `list` command will give you a full list of all commands and their arguments available at the current node:

Example: Typing list at start of OsmoNITB VIEW node prompt

```
OpenBSC> list
  show version
  show online-help
  list
  exit
  help
  enable
```

```
  terminal length <0-512>
  terminal no length
  who
  show history
  show network
  show bts [<0-255>]
  show trx [<0-255>] [<0-255>]
  show timeslot [<0-255>] [<0-255>] [<0-7>]
  show lchan [<0-255>] [<0-255>] [<0-7>] [lchan_nr]
  show lchan summary [<0-255>] [<0-255>] [<0-7>] [lchan_nr]
  show paging [<0-255>]
  show paging-group <0-255> IMSI
  logging enable
  logging disable
  logging filter all (0|1)
  logging color (0|1)
  logging timestamp (0|1)
  logging print extended-timestamp (0|1)
  logging print category (0|1)
  logging set-log-mask MASK
  logging level (all|rll|cc|mm|rr|rsl|nm|mncc|pag|meas|sccp|msc|mgcp|ho|db|ref|gprs|ns| ↩
     bssgp|llc|sndcp|nat|ctrl|smpp|filter|lglobal|llapd|linp|lmux|lmi|lmib|lsms|lctrl|lgtp| ↩
     lstats) (debug|info|notice|error|fatal)
  show logging vty
  show alarms
  show stats
  show stats level (global|peer|subscriber)
  show e1_driver
  show e1_line [line_nr] [stats]
  show e1_timeslot [line_nr] [ts_nr]
  show subscriber (extension|imsi|tmsi|id) ID
  show subscriber cache
  sms send pending
  subscriber create imsi ID
  subscriber (extension|imsi|tmsi|id) ID sms sender (extension|imsi|tmsi|id) SENDER_ID send ↩
      .LINE
  subscriber (extension|imsi|tmsi|id) ID silent-sms sender (extension|imsi|tmsi|id)  ↩
     SENDER_ID send .LINE
  subscriber (extension|imsi|tmsi|id) ID silent-call start (any|tch/f|tch/any|sdcch)
  subscriber (extension|imsi|tmsi|id) ID silent-call stop
  subscriber (extension|imsi|tmsi|id) ID ussd-notify (0|1|2) .TEXT
  subscriber (extension|imsi|tmsi|id) ID update
  show statistics
  show sms-queue
  logging filter imsi IMSI
  show smpp esme
```

Tip

Remember, the list of available commands will change significantly depending on the Osmocom program you are accessing, its software version and the current node you're at. Compare the above example of the OsmoNITB *VIEW* node with the list of the OsmoNITB *TRX* config node:

Example: Typing list at start of OsmoNITB TRX config node prompt

```
OpenBSC(config-net-bts-trx)# list
  help
  list
  write terminal
  write file
```

```
write memory
write
show running-config
exit
end
arfcn <0-1023>
description .TEXT
no description
nominal power <0-100>
max_power_red <0-100>
rsl e1 line E1_LINE timeslot <1-31> sub-slot (0|1|2|3|full)
rsl e1 tei <0-63>
rf_locked (0|1)
timeslot <0-7>
```

7 libosmocore Logging System

In any reasonably complex software it is important to understand how to enable and configure logging in order to get a better insight into what is happening, and to be able to follow the course of action. We therefore ask the reader to bear with us while we explain how the logging subsystem works and how it is configured.

Most Osmocom Software (like `osmo-bts`, `osmo-bsc`, `osmo-nitb`, `osmo-sgsn` and many others) uses the same common logging system.

This chapter describes the architecture and configuration of this common logging system.

The logging system is composed of

- log targets (where to log),
- log categories (who is creating the log line),
- log levels (controlling the verbosity of logging), and
- log filters (filtering or suppressing certain messages).

All logging is done in human-readable ASCII-text. The logging system is configured by means of VTY commands that can either be entered interactively, or read from a configuration file at process start time.

7.1 Log categories

Each sub-system of the program in question typically logs its messages as a different category, allowing fine-grained control over which log messages you will or will not see. For example, in OsmoBSC, there are categories for the protocol layers `rsl`, `rr`, `mm`, `cc` and many others. To get a a list of categories interactively on the vty, type: `logging level ?`

7.2 Log levels

For each of the log categories (see Section 7.1), you can set an independent log level, controlling the level of verbosity. Log levels include:

fatal
Fatal messages, causing abort and/or re-start of a process. This *shouldn't happen*.

error
An actual error has occurred, its cause should be further investigated by the administrator.

notice
A noticeable event has occurred, which is not considered to be an error.

info
Some information about normal/regular system activity is provided.

debug
Verbose information about internal processing of the system, used for debugging purpose. This will log the most.

The log levels are inclusive, e.g. if you select *info*, then this really means that all events with a level of at least *info* will be logged, i.e. including events of *notice*, *error* and *fatal*.

So for example, in OsmoBSC, to set the log level of the Mobility Management category to info, you can use the following command: `log level mm info`.

7.3 Log filters

The default behavior is to filter out everything, i.e. not to log anything. The reason is quite simple: On a busy production setup, logging all events for a given subsystem may very quickly be flooding your console before you have a chance to set a more restrictive filter.

To request no filtering, i.e. see all messages, you may use: `log filter all 1`

In addition to generic filtering, applications can implement special log filters using the same framework to filter on particular context.

For example in OsmoBSC, to only see messages relating to a particular subscriber identified by his IMSI, you may use: `log filter imsi 262020123456789`

7.4 Log targets

Each of the log targets represent certain destination for log messages. It can be configured independently by selecting levels (see Section 7.2) for categories (see Section 7.1) as well as filtering (see Section 7.3) and other options like `logging timestamp` for example.

7.4.1 Logging to the VTY

Logging messages to the interactive command-line interface (VTY) is most useful for occasional investigation by the system administrator.

Logging to the VTY is disabled by default, and needs to be enabled explicitly for each such session. This means that multiple concurrent VTY sessions each have their own logging configuration. Once you close a VTY session, the log target will be destroyed and your log settings be lost. If you re-connect to the VTY, you have to again activate and configure logging, if you wish.

To create a logging target bound to a VTY, you have to use the following command: `logging enable` This doesn't really activate the generation of any output messages yet, it merely creates and attaches a log target to the VTY session. The newly-created target still doesn't have any filter installed, i.e. *all log messages will be suppressed by default*

Next, you can configure the log levels for desired categories in your VTY session. See Section 7.1 for more details on categories and Section 7.2 for the log level details.

For example, to set the log level of the Call Control category to debug, you can use: `log level cc debug`

Finally, after having configured the levels, you still need to set the filter as it's described in Section 7.3.

Tip
If many messages are being logged to a VTY session, it may be hard to impossible to still use the same session for any commands. We therefore recommend to open a second VTY session in parallel, and use one only for logging, while the other is used for interacting with the system. Another option would be to use different log target.

To review the current vty logging configuration, you can use: `show logging vty`

7.4.2 Logging to the ring buffer

To avoid having separate VTY session just for logging output while still having immediate access to them, one can use `alarms` target. It lets you store the log messages inside the ring buffer of a given size which is available with `show alarms` command.

It's configured as follows:

```
OsmoBSC> enable
OsmoBSC# configure terminal
OsmoBSC(config)# log alarms 98
OsmoBSC(config-log)#
```

In the example above 98 is the desired size of the ring buffer (number of messages). Once it's filled, the incoming log messages will push out the oldest messages available in the buffer.

7.4.3 Logging via gsmtap

When debugging complex issues it's handy to be able to reconstruct exact chain of events. This is enabled by using GSMTAP log output where frames sent/received over the air are intersperced with the log lines. It also simplifies the bug handling as users don't have to provide separate .pcap and .log files anymore - everything will be inside self-contained packet dump.

It's configured as follows:

```
OsmoBSC> enable
OsmoBSC# configure terminal
OsmoBSC(config)# log gsmtap 192.168.2.3
OsmoBSC(config-log)#
```

The hostname/ip argument is optional: if omitted the default 127.0.0.1 will be used. The log strings inside GSMTAP are already supported by Wireshark. Capturing for `port 4729` on appropriate interface will reveal log messages including source file name and line number as well as application. This makes it easy to consolidate logs from several different network components alongside the air frames. You can also use Wireshark to quickly filter logs for a given subsystem, severity, file name etc.

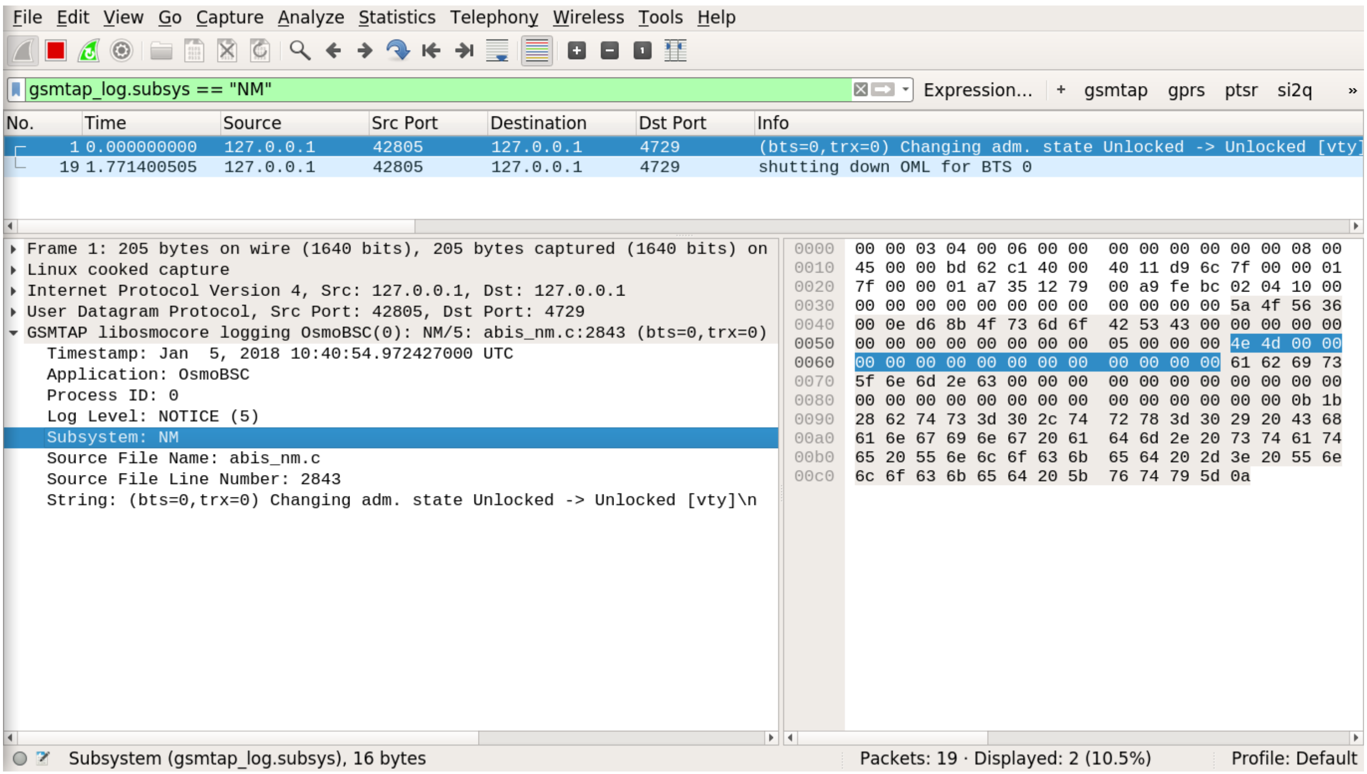

Figure 1: Wireshark with logs delivered over GSMTAP

Note: the logs are also duplicated to stderr when GSMTAP logging is configured because stderr is the default log target which is initialized automatically. To descrease stderr logging to absolute minimum, you can configure it as follows:

```
OsmoBSC> enable
OsmoBSC# configure terminal
OsmoBSC(config)# log stderr
OsmoBSC(config-log)# logging level all fatal
```

7.4.4 Logging to a file

As opposed to Logging to the VTY, logging to files is persistent and stored in the configuration file. As such, it is configured in sub-nodes below the configuration node. There can be any number of log files active, each of them having different settings regarding levels / subsystems.

To configure a new log file, enter the following sequence of commands:

```
OsmoBSC> enable
OsmoBSC# configure terminal
OsmoBSC(config)# log file /path/to/my/file
OsmoBSC(config-log)#
```

This leaves you at the config-log prompt, from where you can set the detailed configuration for this log file. The available commands at this point are identical to configuring logging on the VTY, they include `logging filter`, `logging level` as well as `logging color` and `logging timestamp`.

> **Tip**
> Don't forget to use the `copy running-config startup-config` (or its short-hand `write file`) command to make your logging configuration persistent across application re-start.

> **Note**
> libosmocore currently does not provide file close-and-reopen support by SIGHUP, as used by popular log file rotating solutions. Please contact the Osmocom developers if you require this feature to be implemented.

7.4.5 Logging to syslog

syslog is a standard for computer data logging maintained by the IETF. Unix-like operating systems like GNU/Linux provide several syslog compatible log daemons that receive log messages generated by application programs.

libosmocore based applications can log messages to syslog by using the syslog log target. You can configure syslog logging by issuing the following commands on the VTY:

```
OsmoBSC> enable
OsmoBSC# configure terminal
OsmoBSC(config)# log syslog daemon
OsmoBSC(config-log)#
```

This leaves you at the config-log prompt, from where you can set the detailed configuration for this log file. The available commands at this point are identical to configuring logging on the VTY, they include `logging filter`, `logging level` as well as `logging color` and `logging timestamp`.

> **Note**
> Syslog daemons will normally automatically prefix every message with a time-stamp, so you should disable the libosmocore time-stamping by issuing the `logging timestamp 0` command.

7.4.6 Logging to stderr

If you're not running the respective application as a daemon in the background, you can also use the stderr log target in order to log to the standard error file descriptor of the process.

In order to configure logging to stderr, you can use the following commands:

```
OsmoBSC> enable
OsmoBSC# configure terminal
OsmoBSC(config)# log stderr
OsmoBSC(config-log)#
```

8 Configuring OsmoGGSN

All configuration of OsmoGGSN is performed using the VTY. For more general information on the VTY interface, see Section 6.

8.1 Configuring a virtual GGSN instance

OsmoGGSN can run multiple GGSN instances inside one program/process. Each GGSN instance binds to its own transport-layer GTP IP address and has its own set of APNs and associated IP address pools + tun/gtp devices.

In most usage cases, yo will only have a single GGSN instance inside your configuration file, like in below example:

Example: Single GGSN configuration section

```
ggsn ggsn0
 gtp state-dir /tmp
 gtp bind-ip 127.0.0.6
 apn internet
  gtpu-mode tun
  tun-device tun4
  type-support v4
  ip prefix dynamic 176.16.222.0/24
  ip dns 0 192.168.100.1
  ip dns 1 8.8.8.8
  ip ifconfig 176.16.222.0/24
  no shutdown
```

8.1.1 Creating/Editing a GGSN instance

Creating/Editing a GGSN instance can be done by the following sequence of VTY commands:

```
OsmoGGSN> enable ❶
OsmoGGSN# configure terminal ❷
OsmoGGSN(config)# ggsn ggsn0 ❸
OsmoGGSN(config-ggsn)# ❹
```

❶ Change into privileged mode

❷ Enter the interactive configuration mode

❸ Create or edit the GGSN instance `ggsn0`. The name can be any ASCII string, its significance is only to the local user.

❹ Your prompt is now in the `ggsn` config node, where you can configure the properties of this GGSN instance.

> **NOTE**
> After creating a new GGSN instance, it is in `shutdown` mode. See Section 8.2.4 to take it out of shutdown, but make sure to configure it fully before taking it out of shutdown.

8.1.2 Configuring a GGSN instance

The following two mandatory configuration statements have to be given for every GGSN instance:

```
OsmoGGSN(config-ggsn)# gtp state-dir /var/lib/ggsn/ggsn0 ❶
OsmoGGSN(config-ggsn)# gtp bind-ip 127.0.0.6 ❷
```

❶ Store the GSN restart state in the specified directory

❷ Bind the GGSN instance to the specified local IPv4 address

There are some further configuration statements that can be used at the GGSN node, some examples are given below. For a full list, see the *OsmoGGSN VTY reference manual* [vty-ref-osmoggsn].

```
OsmoGGSN(config-ggsn)# default-apn foobar ❶
```

❶ Configure a default APN to be used if the user-requested APN is not found. The named APN must previously be configured

8.1.3 Deleting a GGSN instance

A GGSN instance can be removed like this

Example: Deleting a GGSN instance

```
OsmoGGSN> enable ❶
OsmoGGSN# configure terminal ❷
OsmoGGSN(config)# no ggsn ggsn0 ❸
```

❶ Change into privileged mode

❷ Enter the interactive configuration mode

❸ Delete the GGSN instance

8.1.4 Taking a GGSN instance out of shutdown

Example: Taking a GGSN instance out of shutdown

```
OsmoGGSN> enable ❶
OsmoGGSN# configure terminal ❷
OsmoGGSN(config)# ggsn ggsn0 ❸
OsmoGGSN(config-ggsn)# no shutdown ggsn ❹
```

❶ Change into privileged mode

❷ Enter the interactive configuration mode

❸ Enter the config ndoe of the GGSN instance `ggsn0`

❹ Take the GGSN instance out of shutdown

8.1.5 Shutting a GGSN instance down

If you would like to take a GGSN instance out of service, you can put it into shutdown mode. This will make the entire GGSN unavailable to user traffic and permit you to e.g. reconfigure it before taking it out of shutdown again.

Example: Shutting down a GGSN instance

```
OsmoGGSN> enable ❶
OsmoGGSN# configure terminal ❷
OsmoGGSN(config)# ggsn ggsn0 ❸
OsmoGGSN(config-ggsn)# shutdown ggsn ❹
```

❶ Change into privileged mode

❷ Enter the interactive configuration mode

❸ Enter the config ndoe of the GGSN instance `ggsn0`

❹ Shut down the GGSN instance

8.2 Configuring an Access Point Name

An Access Point Name (APN) represents a connection to an external packet data network, such as the public Internet or private corporate networsk.

APNs are selected by terminals (MS/UE) when establishing PDP contexts.

Each OsmoGGSN GGSN instance can have any number of APNs configured. Each APN is identified by a string name.

8.2.1 Creating/Editing an APN

Example: Creating a new APN

```
OsmoGGSN> enable ❶
OsmoGGSN# configure terminal ❷
OsmoGGSN(config)# ggsn ggsn0 ❸
OsmoGGSN(config-ggsn)# apn internet ❹
OsmoGGSN(config-ggsn-apn)# ❺
```

❶ Change into privileged mode

❷ Enter the interactive configuration mode

❸ Enter the config node of the GGSN instance `ggsn0`

❹ Create or Edit an APN called `internet`

❺ Your prompt is now in the `ggsn` config node, where you can configure the properties of this GGSN instance.

NOTE
The newly-create APN is created in `shutdown` mode. See Section 8.2.4 to take it out of shutdown.

8.2.2 Configuring an APN

Example: Configuring an APN

```
OsmoGGSN(config-ggsn-apn)# gtpu-mode tun ❶
OsmoGGSN(config-ggsn-apn)# type-support v4 ❷
OsmoGGSN(config-ggsn-apn)# ip prefix dynamic 176.16.222.0/24 ❸
OsmoGGSN(config-ggsn-apn)# ip dns 0 192.168.100.1 ❹
OsmoGGSN(config-ggsn-apn)# ip dns 1 8.8.8.8 ❺
OsmoGGSN(config-ggsn-apn)# ip ifconfig 176.16.222.0/24 ❻
```

❶ Use the userspace GTP-U handling using a TUN device

❷ Support (only) IPv4 Addresses

❸ Specify the pool of dynamic IPv4 addresses to be allocated to PDP contexts

❹ Specify the primary DNS server to be provided using IPCP/PCO

❺ Specify the secondary DNS server to be provided using IPCP/PCO

❻ Request OsmoGGSN to configure the `tun4` device network/netmask

NOTE
If you use the optional `ip ifconfig` command to set the network device address/mask, OsmoGGSN must run with root or `CAP_NET_ADMIN` support. It might be better to configure related tun devices at system startup and run OsmoGGSN as non-privileged user. See Section 8.3 for more details.

8.2.3 Deleting an APN

An APN configuration can be removed like this

Example: Deleting an APN

```
OsmoGGSN> enable ❶
OsmoGGSN# configure terminal ❷
OsmoGGSN(config)# ggsn ggsn0 ❸
OsmoGGSN(config-ggsn)# no apn internet ❹
```

❶ Change into privileged mode

❷ Enter the interactive configuration mode

❸ Enter the config node of the GGSN instance `ggsn0`

❹ Delete the APN `internet`

8.2.4 Taking an APN out of shutdown

In order to bring a deactived APN in `shutdown` state into active operation, use the `no shutdown` command at the APN node as explained in the following example:

Example: Taking an APN out of shutdown

```
OsmoGGSN> enable ❶
OsmoGGSN# configure terminal ❷
OsmoGGSN(config)# ggsn ggsn0 ❸
OsmoGGSN(config-ggsn)# apn internet ❹
OsmoGGSN(config-ggsn-apn)# no shutdown ❺
```

1. Change into privileged mode
2. Enter the interactive configuration mode
3. Enter the config ndoe of the GGSN instance `ggsn0`
4. Enter the config ndoe of the APN `internet`
5. Take the APN out of shutdown

8.2.5 Shutting an APN down

If you would like to take an APN instance out of service, you can put it into shutdown mode. This will make the APN unavailable to user traffic and permit you to e.g. reconfigure it before taking it out of shutdown again.

Example: Shutting down an APN

```
OsmoGGSN> enable ❶
OsmoGGSN# configure terminal ❷
OsmoGGSN(config)# ggsn ggsn0 ❸
OsmoGGSN(config-ggsn)# apn internet ❹
OsmoGGSN(config-ggsn-apn)# shutdown ❺
```

1. Change into privileged mode
2. Enter the interactive configuration mode
3. Enter the config ndoe of the GGSN instance `ggsn0`
4. Enter the config ndoe of the APN `internet`
5. Shut down the APN

8.3 Configuring for running without root privileges

It's possible to run OsmoGGSN without root privileges if the tun devices are already configured.

The interface creation + configuration must then happen before osmo-ggsn starting up. This can be achieved by means such as

- a custom shell script run as root before starting osmo-ggsn (e.g. as init script)
- systemd .netdev and .network files, if your system is using systemd-networkd (see `networkctl status`).

8.3.1 Manual TUN device creation / configuration

If you chose to go for custom shell/init scripts, you may use the `ip` program which is the standard tool for network interface configuration on Linux, part of the `iproute2` package. In order to create a tun device, you must call it like this:

Example: iproute2 command to create a tun device

```
# ip tuntap add dev apn0 mode tun user username group groupname
```

Where *username* and *groupname* correspond to the User and Group that will have ownership over the device, i.e. the privileges which you intend to run osmo-ggsn under, and *apn0* will be the name of the network device created. After creating the interface, you can configure its addresses using standard means like `ip addr add` or your distribution-specific utilities/tools to match the `ip prefix dynamic` config item, and activate the link, for example:

```
# ip addr add 192.168.7.0/24 dev apn0
# ip link set apn0 up
```

8.3.2 systemd based TUN device creation+configuration

If you want to have systemd take care of creating and configuring a tun device for you, you can use the below example config files.

Example: device config via systemd-networkd using apn0.netdev

```
[NetDev]
Name=apn0 ❶
Kind=tun

[Tun]
User=username ❷
Group=username ❸
```

❶ The network interface name of the newly-created device

❷ The username under which you will run OsmoGGSN

❸ The group name under which you will run OsmoGGSN

Example: network settings via systemd-networkd using ggsn.network

```
[Match]
Name=apn0 ❶

[Network]
Address=192.168.7.1 ❷
IPMasquerade=yes ❸
```

❶ The netowrk device name, which must match the one in the apn0.netdev unit file above

❷ The local IP address configured on the device

❸ Requesting systemd to configure IP masquerading for this interface. Depending on your needs, You may not want this if you have proper end-to-end routing set up, and want to have transparent inbound IP access to your GPRS-attached devices.

8.3.3 Config Changes

With the tun device pre-configured in one of the ways outlined above, the main changes in your osmo-ggsn.cfg file are:

- remove `ip ifconfig` directive,
- make sure that `no shutdown` is present in the `apn` section as well as `no shutdown ggsn` in the `ggsn` section.

Example: using externally configured tun device apn0 as non-root

```
ggsn ggsn0
 gtp state-dir /tmp
 gtp bind-ip 127.0.0.6
 apn internet
  gtpu-mode tun
  tun-device apn0
  type-support v4
  ip prefix dynamic 192.168.7.0/24
  ip dns 0 192.168.100.1
  ip dns 1 8.8.8.8
  no shutdown
 default-apn internet
 no shutdown ggsn
```

9 Osmocom Control Interface

The VTY interface as described in Section 6 is aimed at human interaction with the respective Osmocom program.

Other programs **should not** use the VTY interface to interact with the Osmocom software, as parsing the textual representation is cumbersome, inefficient, and will break every time the formatting is changed by the Osmocom developers.

Instead, the *Control Interface* was introduced as a programmatic interface that can be used to interact with the respective program.

9.1 Control Interface Protocol

The control interface protocol is a mixture of binary framing with text based payload.

The protocol for the control interface is wrapped inside the IPA multiplex header with the stream identifier set to IPAC_PROTO_OSMO (0xEE).

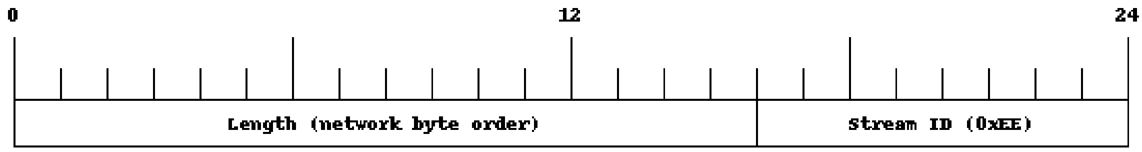

Figure 2: IPA header for control protocol

Inside the IPA header is a single byte of extension header with protocol ID 0x00 which indicates the control interface.

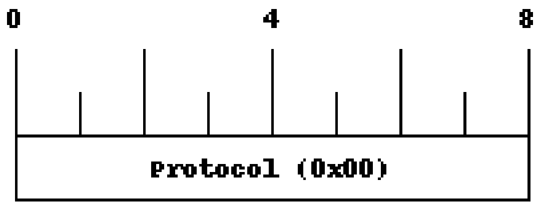

Figure 3: IPA extension header for control protocol

After the concatenation of the two above headers, the plain-text payload message starts. The format of that plain text is illustrated for each operation in the respective message sequence chart in the chapters below.

The fields specified below follow the following meaning:

<id>
: A numeric identifier, uniquely identifying this particular operation. `0` is not allowed. It will be echoed back in any response to a particular request.

<var>
The name of the variable / field affected by the GET / SET / TRAP operation. Which variables/fields are available is dependent on the specific application under control.

<val>
The value of the variable / field

<reason>
A text formatted, human-readable reason why the operation resulted in an error.

9.1.1 GET operation

The GET operation is performed by an external application to get a certain value from inside the Osmocom application.

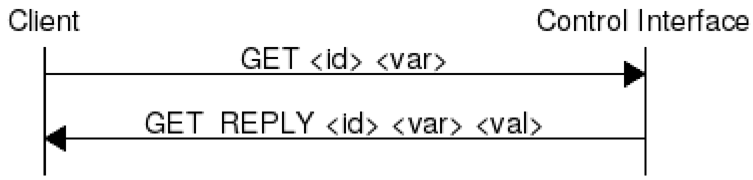

Figure 4: Control Interface GET operation (successful outcome)

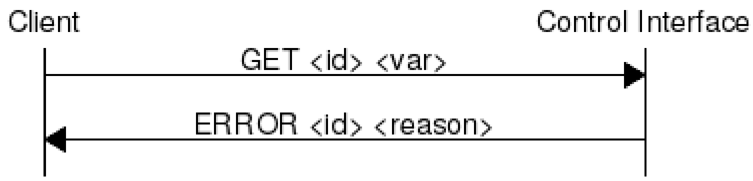

Figure 5: Control Interface GET operation (unsuccessful outcome)

9.1.2 SET operation

The SET operation is performed by an external application to set a value inside the Osmocom application.

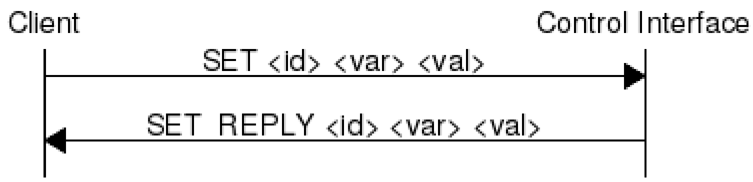

Figure 6: Control Interface SET operation (successful outcome)

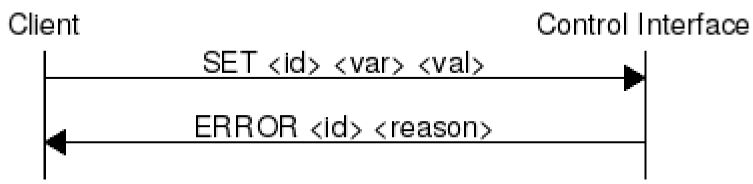

Figure 7: Control Interface SET operation (unsuccessful outcome)

9.1.3 TRAP operation

The program can at any time issue a trap. The term is used in the spirit of SNMP.

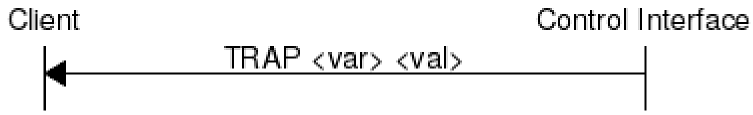

Figure 8: Control Interface TRAP operation

9.2 Common variables

There are several variables which are common to all the programs using control interface. They are described in the following table.

Table 2: Variables available over control interface

Name	Access	Value	Comment
counter.*	RO		Get counter value.
rate_ctr.*	RO		Get list of rate counter groups.
rate_ctr.IN.GN.GI.name	RO		Get value for interval IN of rate counter name which belong to group named GN with index GI.

Those read-only variables allow to get value of arbitrary counter using its name.

For example "`rate_ctr.per_hour.bsc.0.handover:timeout`" is the number of handover timeouts per hour.

Of course for that to work the program in question have to register corresponding counter names and groups using libosmocore functions.

In the example above, "`bsc`" is the rate counter group name and "`0`" is its index. It is possible to obtain all the rate counters in a given group by requesting "`rate_ctr.per_sec.bsc.*`" variable.

The list of available groups can be obtained by requesting "`rate_ctr.*`" variable.

The rate counter group name have to be prefixed with interval specification which can be any of "**per_sec**", "**per_min**", "**per_hour**", "**per_day**" or "**abs**" for absolute value.

The old-style counters available via "`counter.*`" variables are superceeded by "`rate_ctr.abs`" so its use is discouraged. There might still be some applications not yet converted to rate_ctr.

9.3 Control Interface python examples

In the `osmo-python-tests` repository, there is an example python script called `scripts/osmo_ctrl.py` which implements the Osmocom control interface protocol.

You can use this tool either stand-alone to perform control interface operations against an Osmocom program, or you can use it as a reference for developing your own python software talking to the control interface.

Another implementation is in `scripts/osmo_rate_ctr2csv.py` which will retrieve performance counters for a given Osmocom program and output it in csv format. This can be used to periodically (using systemd timer for example) retrieve data to build KPI and evaluate how it changes over time.

Internally it uses "rate_ctr.*" variable described in [?] to get the list of counter groups and than request all the counters in each group. Applications interested in individual metrics can request it directly using rate_ctr2csv.py as an example.

9.3.1 Getting rate counters

Example: Use rate_ctr2csv.py to get rate counters from OsmoBSC

```
$ ./scripts/osmo_rate_ctr2csv.py --header
Connecting to localhost:4249...
Getting rate counter groups info...
"group","counter","absolute","second","minute","hour","day"
"e1inp.0","hdlc:abort","0","0","0","0","0"
"e1inp.0","hdlc:bad_fcs","0","0","0","0","0"
"e1inp.0","hdlc:overrun","0","0","0","0","0"
"e1inp.0","alarm","0","0","0","0","0"
"e1inp.0","removed","0","0","0","0","0"
"bsc.0","chreq:total","0","0","0","0","0"
"bsc.0","chreq:no_channel","0","0","0","0","0"
...
"msc.0","call:active","0","0","0","0","0"
"msc.0","call:complete","0","0","0","0","0"
"msc.0","call:incomplete","0","0","0","0","0"
Completed: 44 counters from 3 groups received.
```

9.3.2 Setting a value

Example: Use osmo_ctrl.py to set the short network name of OsmoBSC

```
$ ./osmo_ctrl.py -d localhost -s short-name 32C3
Got message: SET_REPLY 1 short-name 32C3
```

9.3.3 Getting a value

Example: Use osmo_ctrl.py to get the mnc of OsmoBSC

```
$ ./osmo_ctrl.py -d localhost -g mnc
Got message: GET_REPLY 1 mnc 262
```

9.3.4 Listening for traps

You can use osmo_ctrl.py to listen for traps the following way:

Example: Using osmo_ctrl.py to listen for traps:

```
$ ./osmo_ctrl.py -d localhost -m
❶
```

❶ the command will not return and wait for any TRAP messages to arrive

10 Glossary

2FF
2nd Generation Form Factor; the so-called plug-in SIM form factor

3FF
3rd Generation Form Factor; the so-called microSIM form factor

3GPP
3rd Generation Partnership Project

4FF
4th Generation Form Factor; the so-called nanoSIM form factor

A Interface
Interface between BTS and BSC, traditionally over E1 (*3GPP TS 48.008* [3gpp-ts-48-008])

A3/A8
Algorithm 3 and 8; Authentication and key generation algorithm in GSM and GPRS, typically COMP128v1/v2/v3 or MILENAGE are typically used

A5
Algorithm 5; Air-interface encryption of GSM; currently only A5/0 (no encryption), A5/1 and A5/3 are in use

Abis Interface
Interface between BTS and BSC, traditionally over E1 (*3GPP TS 48.058* [3gpp-ts-48-058] and *3GPP TS 52.021* [3gpp-ts-52-021])

ACC
Access Control Class; every BTS broadcasts a bit-mask of permitted ACC, and only subscribers with a SIM of matching ACC are permitted to use that BTS

AGCH
Access Grant Channel on Um interface; used to assign a dedicated channel in response to RACH request

AGPL
GNU Affero General Public License, a copyleft-style Free Software License

ARFCN
Absolute Radio Frequency Channel Number; specifies a tuple of uplink and downlink frequencies

AUC
Authentication Center; central database of authentication key material for each subscriber

BCCH
Broadcast Control Channel on Um interface; used to broadcast information about Cell and its neighbors

BCC
Base Station Color Code; short identifier of BTS, lower part of BSIC

BTS
Base Transceiver Station

BSC
Base Station Controller

BSIC
Base Station Identity Code; 16bit identifier of BTS within location area

BSSGP
Base Station Subsystem Gateway Protocol (*3GPP TS 48.018* [3gpp-ts-48-018])

BVCI
BSSGP Virtual Circuit Identifier

CBCH
Cell Broadcast Channel; used to transmit Cell Broadcast SMS (SMS-CB)

CC
Call Control; Part of the GSM Layer 3 Protocol

CCCH
Common Control Channel on Um interface; consists of RACH (uplink), BCCH, PCH, AGCH (all downlink)

Cell
A cell in a cellular network, served by a BTS

CEPT
Conférence européenne des administrations des postes et des télécommunications; European Conference of Postal and Telecommunications Administrations.

CGI
Cell Global Identifier comprised of MCC, MNC, LAC and BSIC

dB
deci-Bel; relative logarithmic unit

dBm
deci-Bel (milliwatt); unit of measurement for signal strength of radio signals

DHCP
Dynamic Host Configuration Protocol (*IETF RFC 2131* [ietf-rfc2131]

downlink
Direction of messages / signals from the network core towards the mobile phone

DSP
Digital Signal Processor

dvnixload
Tool to program UBL and the Bootloader on a sysmoBTS

EDGE
Enhanced Data rates for GPRS Evolution; Higher-speed improvement of GPRS; introduces 8PSK

EGPRS
Enhanced GPRS; the part of EDGE relating to GPRS services

ESME
External SMS Entity; an external application interfacing with a SMSC over SMPP

ETSI
European Telecommunications Standardization Institute

FPGA
Field Programmable Gate Array; programmable digital logic hardware

Gb
Interface between PCU and SGSN in GPRS/EDGE network; uses NS, BSSGP, LLC

GERAN
GPRS/EDGE Radio Access Network

GGSN
GPRS Gateway Support Node; gateway between GPRS and external (IP) network

GMSK
: Gaussian Minimum Shift Keying; modulation used for GSM and GPRS

GPL
: GNU General Public License, a copyleft-style Free Software License

Gp
: Gp interface between SGSN and GGSN; uses GTP protocol

GPS
: Global Positioning System; provides a highly accurate clock reference besides the global position

GSM
: Global System for Mobile Communications. ETSI/3GPP Standard of a 2G digital cellular network

GSMTAP
: GSM tap; pseudo standard for encapsulating GSM protocol layers over UDP/IP for analysis

GT
: Global Title; an address in SCCP

GTP
: GPRS Tunnel Protocol; used between SGSN and GGSN

HLR
: Home Location Register; central subscriber database of a GSM network

HPLMN
: Home PLMN; the network that has issued the subscriber SIM and has his record in HLR

IE
: Information Element

IMEI
: International Mobile Equipment Identity; unique identifier for the mobile phone

IMSI
: International Mobile Subscriber Identity; 15-digit unique identifier for the subscriber/SIM; starts with MCC/MNC of issuing operator

IP
: Internet Protocol (*IETF RFC 791* [?])

IPA
: *ip.access GSM over IP* protocol; used to multiplex a single TCP connection

LAC
: Location Area Code; 16bit identifier of Location Area within network

LAPD
: Link Access Protocol, D-Channel (*ITU-T Q.921* [itu-t-q921])

LAPDm
: Link Access Protocol Mobile (*3GPP TS 44.006* [3gpp-ts-44-006])

LLC
: Logical Link Control; GPRS protocol between MS and SGSN (*3GPP TS 44.064* [3gpp-ts-44-064])

Location Area
: Location Area; a geographic area containing multiple BTS

M2PA
: MTP2 Peer-to-Peer Adaptation; a SIGTRAN Variant (*RFC 4165* [ietf-rfc4165])

M2UA
MTP2 User Adaptation; a SIGTRAN Variant (*RFC 3331* [ietf-rfc3331])

M3UA
MTP3 User Adaptation; a SIGTRAN Variant (*RFC 4666* [ietf-rfc4666])

MCC
Mobile Country Code; unique identifier of a country, e.g. 262 for Germany

MFF
Machine-to-Machine Form Factor; a SIM chip package that is soldered permanently onto M2M device circuit boards.

MGW
Media Gateway

MM
Mobility Management; part of the GSM Layer 3 Protocol

MNC
Mobile Network Code; identifies network within a country; assigned by national regulator

MNO
Mobile Network Operator; operator with physical radio network under his MCC/MNC

MS
Mobile Station; a mobile phone / GSM Modem

MSC
Mobile Switching Center; network element in the circuit-switched core network

MSISDN
Mobile Subscriber ISDN Number; telephone number of the subscriber

MTP
Message Transfer Part; SS7 signaling protocol (*ITU-T Q.701* [itu-t-q701])

MVNO
Mobile Virtual Network Operator; Operator without physical radio network

NCC
Network Color Code; assigned by national regulator

NITB
Network In The Box; combines functionality traditionally provided by BSC, MSC, VLR, HLR, SMSC functions; see OsmoNITB

NSEI
NS Entity Identifier

NVCI
NS Virtual Circuit Identifier

NWL
Network Listen; ability of some BTS to receive downlink from other BTSs

NS
Network Service; protocol on Gb interface (*3GPP TS 48.016* [3gpp-ts-48-016])

OCXO
Oven Controlled Crystal Oscillator; very high precision oscillator, superior to a VCTCXO

OML
Operation & Maintenance Link (ETSI/*3GPP TS 52.021* [3gpp-ts-52-021])

OpenBSC
: Open Source implementation of GSM network elements, specifically OsmoBSC, OsmoNITB, OsmoSGSN

OpenGGSN
: Open Source implementation of a GPRS Packet Control Unit

OpenVPN
: Open-Source Virtual Private Network; software employed to establish encrypted private networks over untrusted public networks

Osmocom
: Open Source MObile COMmunications; collaborative community for implementing communications protocols and systems, including GSM, GPRS, TETRA, DECT, GMR and others

OsmoBSC
: Open Source implementation of a GSM Base Station Controller

OsmoNITB
: Open Source implementation of a GSM Network In The Box, combines functionality traditionally provided by BSC, MSC, VLR, HLR, AUC, SMSC

OsmoSGSN
: Open Source implementation of a Serving GPRS Support Node

OsmoPCU
: Open Source implementation of a GPRS Packet Control Unit

OTA
: Over-The-Air; Capability of operators to remotely reconfigure/reprogram ISM/USIM cards

PC
: Point Code; an address in MTP

PCH
: Paging Channel on downlink Um interface; used by network to page an MS

PCU
: Packet Control Unit; used to manage Layer 2 of the GPRS radio interface

PDCH
: Packet Data Channel on Um interface; used for GPRS/EDGE signalling + user data

PIN
: Personal Identification Number; a number by which the user authenticates to a SIM/USIM or other smart card

PLMN
: Public Land Mobile Network; specification language for a single GSM network

PUK
: PIN Unblocking Code; used to unblock a blocked PIN (after too many wrong PIN attempts)

RAC
: Routing Area Code; 16bit identifier for a Routing Area within a Location Area

RACH
: Random Access Channel on uplink Um interface; used by MS to request establishment of a dedicated channel

RAM
: Remote Application Management; Ability to remotely manage (install, remove) Java Applications on SIM/USIM Card

RF
: Radio Frequency

RFM
Remote File Management; Ability to remotely manage (write, read) files on a SIM/USIM card

Roaming
Procedure in which a subscriber of one network is using the radio network of another network, often in different countries; in some countries national roaming exists

Routing Area
Routing Area; GPRS specific sub-division of Location Area

RR
Radio Resources; Part of the GSM Layer 3 Protocol

RSL
Radio Signalling Link (*3GPP TS 48.058* [3gpp-ts-48-058])

RTP
Real-Time Transport Protocol (*IETF RFC 3550* [ietf-rfc3550]); Used to transport audio/video streams over UDP/IP

SACCH
Slow Associate Control Channel on Um interface; bundled to a TCH or SDCCH, used for signalling in parallel to active dedicated channel

SCCP
Signaling Connection Control Part; SS7 signaling protocol (*ITU-T Q.711* [itu-t-q711])

SDCCH
Slow Dedicated Control Channel on Um interface; used for signalling and SMS transport in GSM

SDK
Software Development Kit

SIGTRAN
Signaling Transport over IP (*IETF RFC 2719* [ietf-rfc2719])

SIM
Subscriber Identity Module; small chip card storing subscriber identity

Site
A site is a location where one or more BTSs are installed, typically three BTSs for three sectors

SMPP
Short Message Peer-to-Peer; TCP based protocol to interface external entities with an SMSC

SMSC
Short Message Service Center; store-and-forward relay for short messages

SS7
Signaling System No. 7; Classic digital telephony signaling system

SSH
Secure Shell; *IETF RFC 4250* [ietf-rfc4251] to 4254

SSN
Sub-System Number; identifies a given SCCP Service such as MSC, HLR

STP
Signaling Transfer Point; A Router in SS7 Networks

SUA
SCCP User Adaptation; a SIGTRAN Variant (*RFC 3868* [ietf-rfc3868])

syslog
System logging service of UNIX-like operating systems

System Information
: A set of downlink messages on the BCCH and SACCH of the Um interface describing properties of the cell and network

TCH
: Traffic Channel; used for circuit-switched user traffic (mostly voice) in GSM

TCP
: Transmission Control Protocol; (*IETF RFC 793* [ietf-rfc793])

TFTP
: Trivial File Transfer Protocol; (*IETF RFC 1350* [ietf-rfc1350])

TRX
: Transceiver; element of a BTS serving a single carrier

u-Boot
: Boot loader used in various embedded systems

UBI
: An MTD wear leveling system to deal with NAND flash in Linux

UBL
: Initial bootloader loaded by the TI Davinci SoC

UDP
: User Datagram Protocol (*IETF RFC 768* [ietf-rfc768])

UICC
: Universal Integrated Chip Card; A smart card according to *ETSI TR 102 216* [etsi-tr102216]

Um interface
: U mobile; Radio interface between MS and BTS

uplink
: Direction of messages: Signals from the mobile phone towards the network

USIM
: Universal Subscriber Identity Module; application running on a UICC to provide subscriber identity for UMTS and GSM networks

VCTCXO
: Voltage Controlled, Temperature Compensated Crystal Oscillator; a precision oscillator, superior to a classic crystal oscillator, but inferior to an OCXO

VPLMN
: Visited PLMN; the network in which the subscriber is currently registered; may differ from HPLMN when on roaming

VTY
: Virtual TeletYpe; a textual command-line interface for configuration and introspection, e.g. the OsmoBSC configuration file as well as its telnet link on port 4242

A Osmocom TCP/UDP Port Numbers

The Osmocom GSM system utilizes a variety of TCP/IP based protocols. The table below provides a reference as to which port numbers are used by which protocol / interface.

Table 3: TCP/UDP port numbers

L4 Protocol	Port Number	Purpose	Software
UDP	2427	MGCP GW	osmo-bsc_mgcp, osmo-mgw
TCP	2775	SMPP (SMS interface for external programs)	osmo-nitb
TCP	3002	A-bis/IP OML	osmo-bts, osmo-bsc, osmo-nitb
TCP	3003	A-bis/IP RSL	osmo-bts, osmo-bsc, osmo-nitb
TCP	4239	telnet (VTY)	osmo-stp
TCP	4240	telnet (VTY)	osmo-pcu
TCP	4241	telnet (VTY)	osmo-bts
TCP	4242	telnet (VTY)	osmo-nitb, osmo-bsc, cellmgr-ng
TCP	4243	telnet (VTY)	osmo-bsc_mgcp, osmo-mgw
TCP	4244	telnet (VTY)	osmo-bsc_nat
TCP	4245	telnet (VTY)	osmo-sgsn
TCP	4246	telnet (VTY)	osmo-gbproxy
TCP	4247	telnet (VTY)	OsmocomBB
TCP	4249	Control Interface	osmo-nitb, osmo-bsc
TCP	4250	Control Interface	osmo-bsc_nat
TCP	4251	Control Interface	osmo-sgsn
TCP	4252	telnet (VTY)	sysmobts-mgr
TCP	4253	telnet (VTY)	osmo-gtphub
TCP	4254	telnet (VTY)	osmo-msc
TCP	4255	Control Interface	osmo-msc
TCP	4256	telnet (VTY)	osmo-sip-connector
TCP	4257	Control Interface	osmo-ggsn, ggsn (OpenGGSN)
TCP	4258	telnet (VTY)	osmo-hlr
TCP	4259	Control Interface	osmo-hlr
TCP	4260	telnet (VTY)	osmo-ggsn
TCP	4261	telnet (VTY)	osmo-hnbgw
TCP	4262	Control Interface	osmo-hnbgw
UDP	4729	GSMTAP	Almost every osmocom project
TCP	5000	A/IP	osmo-bsc, osmo-bsc_nat
UDP	2427	GSMTAP	osmo-pcu, osmo-bts
UDP	23000	GPRS-NS over IP default port	osmo-pcu, osmo-sgsn, osmo-gbproxy

B Bibliography / References

B.0.4.0.1 References

[1] [osmobts-abis-spec] Neels Hofmeyr & Harald Welte. OsmoBTS Abis Protocol Specification. http://ftp.osmocom.org/docs/latest/osmobts-abis.pdf

[2] [userman-osmobts] Osmocom Project: OsmoBTS User Manual. http://ftp.osmocom.org/docs/latest/osmobts-usermanual.pdf

[3] [vty-ref-osmobts] Osmocom Project: OsmoBTS VTY Reference Manual. http://ftp.osmocom.org/docs/latest/-osmobts-vty-reference.pdf

[4] [userman-osmobsc] Osmocom Project: OsmoBSC User Manual. http://ftp.osmocom.org/docs/latest/osmobsc-usermanual.pdf

[5] [vty-ref-osmobsc] Osmocom Project: OsmoBSC VTY Reference Manual. http://ftp.osmocom.org/docs/latest/-osmobsc-vty-reference.pdf

[6] [userman-osmomsc] Osmocom Project: OsmoMSC User Manual. http://ftp.osmocom.org/docs/latest/-osmomsc-usermanual.pdf

[7] [vty-ref-osmomsc] Osmocom Project: OsmoMSC VTY Reference Manual. http://ftp.osmocom.org/docs/-latest/osmomsc-vty-reference.pdf

[8] [userman-osmohlr] Osmocom Project: OsmoHLR User Manual. http://ftp.osmocom.org/docs/latest/osmohlr-usermanual.pdf

[9] [vty-ref-osmohlr] Osmocom Project: OsmoHLR VTY Reference Manual. http://ftp.osmocom.org/docs/latest/-osmohlr-vty-reference.pdf

[10] [userman-osmopcu] Osmocom Project: OsmoPCU User Manual. http://ftp.osmocom.org/docs/latest/osmopcu-usermanual.pdf

[11] [vty-ref-osmopcu] Osmocom Project: OsmoPCU VTY Reference Manual. http://ftp.osmocom.org/docs/latest/-osmopcu-vty-reference.pdf

[12] [userman-osmonitb] Osmocom Project: OsmoNITB User Manual. http://ftp.osmocom.org/docs/latest/-osmonitb-usermanual.pdf

[13] [vty-ref-osmonitb] Osmocom Project: OsmoNITB VTY Reference Manual. http://ftp.osmocom.org/docs/-latest/osmonitb-vty-reference.pdf

[14] [userman-osmosgsn] Osmocom Project: OsmoSGSN User Manual. http://ftp.osmocom.org/docs/latest/-osmosgsn-usermanual.pdf

[15] [vty-ref-osmosgsn] Osmocom Project: OsmoSGSN VTY Reference Manual. http://ftp.osmocom.org/docs/-latest/osmonitb-vty-reference.pdf

[16] [userman-osmoggsn] Osmocom Project: OpenGGSN User Manual. http://ftp.osmocom.org/docs/latest/-osmoggsn-usermanual.pdf

[17] [vty-ref-osmoggsn] Osmocom Project: OsmoGGSN VTY Reference Manual. http://ftp.osmocom.org/docs/-latest/osmoggsn-vty-reference.pdf

[18] [3gpp-ts-23-048] 3GPP TS 23.048: Security mechanisms for the (U)SIM application toolkit; Stage 2 http://www.3gpp.org/DynaReport/23048.htm

[19] [3gpp-ts-24-007] 3GPP TS 24.007: Mobile radio interface signalling layer 3; General Aspects http://www.3gpp.org/DynaReport/24007.htm

[20] [3gpp-ts-24-008] 3GPP TS 24.008: Mobile radio interface Layer 3 specification; Core network protocols; Stage 3. http://www.3gpp.org/dynareport/24008.htm

[21] [3gpp-ts-31-101] 3GPP TS 31.101: UICC-terminal interface; Physical and logical characteristics http://www.3gpp.org/DynaReport/31101.htm

[22] [3gpp-ts-31-102] 3GPP TS 31.102: Characteristics of the Universal Subscriber Identity Module (USIM) application http://www.3gpp.org/DynaReport/31102.htm

[23] [3gpp-ts-31-111] 3GPP TS 31.111: Universal Subscriber Identity Module (USIM) Application Toolkit (USAT) http://www.3gpp.org/DynaReport/31111.htm

[24] [3gpp-ts-31-115] 3GPP TS 31.115: Secured packet structure for (Universal) Subscriber Identity Module (U)SIM Toolkit applications http://www.3gpp.org/DynaReport/31115.htm

[25] [3gpp-ts-31-116] 3GPP TS 31.116: Remote APDU Structure for (U)SIM Toolkit applications http://www.3gpp.org/DynaReport/31116.htm

[26] [3gpp-ts-35-205] 3GPP TS 35.205: 3G Security; Specification of the MILENAGE algorithm set: General

[27] [3gpp-ts-35-206] 3GPP TS 35.206: 3G Security; Specification of the MILENAGE algorithm set: Algorithm specification http://www.3gpp.org/DynaReport/35206.htm

[28] [3gpp-ts-44-006] 3GPP TS 44.006: Mobile Station - Base Station System (MS - BSS) interface; Data Link (DL) layer specification http://www.3gpp.org/DynaReport/44006.htm

[29] [3gpp-ts-44-064] 3GPP TS 44.064: Mobile Station - Serving GPRS Support Node (MS-SGSN); Logical Link Control (LLC) Layer Specification http://www.3gpp.org/DynaReport/44064.htm

[30] [3gpp-ts-48-008] 3GPP TS 48.008: Mobile Switching Centre - Base Station system (MSC-BSS) interface; Layer 3 specification http://www.3gpp.org/DynaReport/48008.htm

[31] [3gpp-ts-48-016] 3GPP TS 48.016: General Packet Radio Service (GPRS); Base Station System (BSS) - Serving GPRS Support Node (SGSN) interface; Network service http://www.3gpp.org/DynaReport/48016.htm

[32] [3gpp-ts-48-018] 3GPP TS 48.018: General Packet Radio Service (GPRS); Base Station System (BSS) - Serving GPRS Support Node (SGSN); BSS GPRS protocol (BSSGP) http://www.3gpp.org/DynaReport/48018.htm

[33] [3gpp-ts-48-056] 3GPP TS 48.056: Base Station Controller - Base Transceiver Station (BSC - BTS) interface; Layer 2 specification http://www.3gpp.org/DynaReport/48056.htm

[34] [3gpp-ts-48-058] 3GPP TS 48.058: Base Station Controller - Base Transceiver Station (BSC - BTS) Interface; Layer 3 specification http://www.3gpp.org/DynaReport/48058.htm

[35] [3gpp-ts-51-011] 3GPP TS 51.011: Specification of the Subscriber Identity Module - Mobile Equipment (SIM-ME) interface

[36] [3gpp-ts-51-014] 3GPP TS 51.014: Specification of the SIM Application Toolkit for the Subscriber Identity Module - Mobile Equipment (SIM - ME) interface http://www.3gpp.org/DynaReport/51014.htm

[37] [3gpp-ts-52-021] 3GPP TS 52.021: Network Management (NM) procedures and messages on the A-bis interface http://www.3gpp.org/DynaReport/52021.htm

[38] [etsi-tr102216] ETSI TR 102 216: Smart cards http://www.etsi.org/deliver/etsi_tr/102200_102299/102216/-03.00.00_60/tr_102216v030000p.pdf

[39] [etsi-ts102221] ETSI TS 102 221: Smart Cards; UICC-Terminal interface; Physical and logical characteristics http://www.etsi.org/deliver/etsi_ts/102200_102299/102221/13.01.00_60/ts_102221v130100p.pdf

[40] [etsi-ts101220] ETSI TS 101 220: Smart Cards; ETSI numbering system for telecommunication application providers http://www.etsi.org/deliver/etsi_ts/101200_101299/101220/12.00.00_60/ts_101220v120000p.pdf

[41] [ietf-rfc768] IETF RFC 768: Internet Protocol https://tools.ietf.org/html/rfc791

[42] [ietf-rfc793] IETF RFC 793: Transmission Control Protocol https://tools.ietf.org/html/rfc793

[43] [ietf-rfc1350] IETF RFC 1350: Trivial File Transfer Protool https://tools.ietf.org/html/rfc1350

[44] [ietf-rfc2131] IETF RFC 2131: Dynamic Host Configuration Protocol https://tools.ietf.org/html/rfc2131

[45] [ietf-rfc2719] IETF RFC 2719: Signal Transport over IP https://tools.ietf.org/html/rfc2719

[46] [ietf-rfc3331] IETF RFC 3331: Message Transfer Part 2 User Adaptation Layer https://tools.ietf.org/html/-rfc3331

[47] [ietf-rfc3550] IETF RFC 3550: RTP: A Transport protocol for Real-Time Applications https://tools.ietf.org/-html/rfc3550

[48] [ietf-rfc3868] IETF RFC 3868: SCCP User Adaptation Layer https://tools.ietf.org/html/rfc3868

[49] [ietf-rfc4165] IETF RFC 4165: Message Transfer Part 2 Peer-to-Peeer Adaptation Layer https://tools.ietf.org/-html/rfc4165

[50] [ietf-rfc4251] IETF RFC 4251: The Secure Shell (SSH) Protocol Architecture https://tools.ietf.org/html/-rfc4251

[51] [ietf-rfc4666] IETF RFC 4666: Message Transfer Part 3 User Adaptation Layer https://tools.ietf.org/html/-rfc4666

[52] [itu-t-q701] ITU-T Q.701: Functional Description of the Message Transfer Part (MTP) https://www.itu.int/rec/-T-REC-Q.701/en/

[53] [itu-t-q711] ITU-T Q.711: Functional Description of the Signalling Connection Control Part https://www.itu.int/rec/T-REC-Q.711/en/

[54] [itu-t-q713] ITU-T Q.713: Signalling connection control part formats and codes https://www.itu.int/rec/T-REC-Q.713/en/

[55] [itu-t-q714] ITU-T Q.714: Signalling connection control part procedures https://www.itu.int/rec/T-REC-Q.714/en/

[56] [itu-t-q921] ITU-T Q.921: ISDN user-network interface - Data link layer specification https://www.itu.int/rec/-T-REC-Q.921/en

[57] [smpp-34] SMPP Develoepers Forum. Short Message Peer-to-Peer Protocol Specification v3.4 http://docs.nimta.com/SMPP_v3_4_Issue1_2.pdf

[58] [gnu-agplv3] Free Software Foundation. GNU Affero General Public License. http://www.gnu.org/licenses/-agpl-3.0.en.html

C GNU Free Documentation License

Version 1.3, 3 November 2008

C.1 PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document “free” in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of “copyleft”, which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

C.2 APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The “Document”, below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as “you”. You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A “Modified Version” of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A “Secondary Section” is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document’s overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The “Invariant Sections” are certain Secondary Section whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The “Cover Texts” are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A “Transparent” copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not Transparent is called “Opaque”.

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The “Title Page” means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, Title Page means the text near the most prominent appearance of the work’s title, preceding the beginning of the body of the text.

The “publisher” means any person or entity that distributes copies of the Document to the public.

A section “Entitled XYZ” means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as “Acknowledgements”, “Dedications”, “Endorsements”, or “History”.) To “Preserve the Title” of such a section when you modify the Document means that it remains a section “Entitled XYZ” according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

C.3 VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section Section C.4.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

C.4 COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document’s license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

C.5 MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

a. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

b. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

c. State on the Title Page the name of the publisher of the Modified Version, as the publisher.

d. Preserve all the copyright notices of the Document.

e. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

f. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

g. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

h. Include an unaltered copy of this License.

i. Preserve the section Entitled “History”, Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled “History” in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

j. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the “History” section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

k. For any section Entitled “Acknowledgements” or “Dedications”, Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

l. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

m. Delete any section Entitled “Endorsements”. Such a section may not be included in the [?].

n. Do not retitle any existing section to be Entitled “Endorsements” or to conflict in title with any Invariant Sections.

o. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Section and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version’s license notice. These titles must be distinct from any other section titles.

You may add a section Entitled “Endorsements”, provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

C.6 COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled “History” in the various original documents, forming one section Entitled “History”; likewise combine any sections Entitled “Acknowledgements”, and any sections Entitled “Dedications”. You must delete all sections Entitled “Endorsements”.

C.7 COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

C.8 AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an “aggregate” if the copyright resulting from the compilation is not used to limit the legal rights of the compilation’s users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

C.9 TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

C.10 TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

C.11 FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

C.12 RELICENSING

"Massive Multiauthor Collaboration Site" (or "MMC Site") means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A "Massive Multiauthor Collaboration" (or "MMC") contained in the site means any set of copyrightable works thus published on the MMC site.

"CC-BY-SA" means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

“Incorporate” means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is “eligible for relicensing” if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

C.13 ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

```
Copyright (c)  YEAR  YOUR NAME.
Permission is granted to copy, distribute and/or modify this document
under the terms of the GNU Free Documentation License, Version 1.3
or any later version published by the Free Software Foundation;
with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.
A copy of the license is included in the section entitled ``GNU
Free Documentation License''.
```

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the “with… Texts.” line with this:

```
with the Invariant Sections being LIST THEIR TITLES, with the
Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.
```

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.