sysmocom

sysmocom - s.f.m.c. GmbH

OsmoNITB User Manual

by Holger Freyther and 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 13, 2012	Initial version.	HF
2	February 2016	Conversion to asciidoc, removal of sysmoBTS specific parts.	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 into what is today known as 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 quick also commercial interest, contribution and adoption. This added 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.

It has been a most exciting ride during the last seven years. I wouldn't want to miss it under any circumstances.

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

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.

1.2 Endorsements

This version of the manual is endorsed by Harald Welte as the official version of the manual.

While the GFDL license (see Appendix C) permits anyone to create and distribute modified versions of this manual, such modified versions must remove the above endorsement.

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 as a company to provide products and services related to Osmocom.

sysmocom and its staff have by far contributed 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 operates in licensed spectrum, please always double-check that you have all required licenses and that you do not transmit on any ARFCN 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.

2.4.5 Documentation License

Please see Appendix C for further information.

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

This manual should help you getting started with OsmoNITB. It will cover aspects of configuring and running the OsmoNITB.

4.1 About OsmoNITB

OsmoNITB is one particular version of the OpenBSC software suite. Unlike classic, distributed, hierarchical GSM networks, OsmoNITB implements all parts of a GSM Network (BSC, MSC, VLR, HLR, AUC, SMSC) *in the box*, i.e. in one element.

The difference between classic GSM network architecture and the OsmoNITB based GSM network architecture is illustrated in Figure 1 and Figure 2.

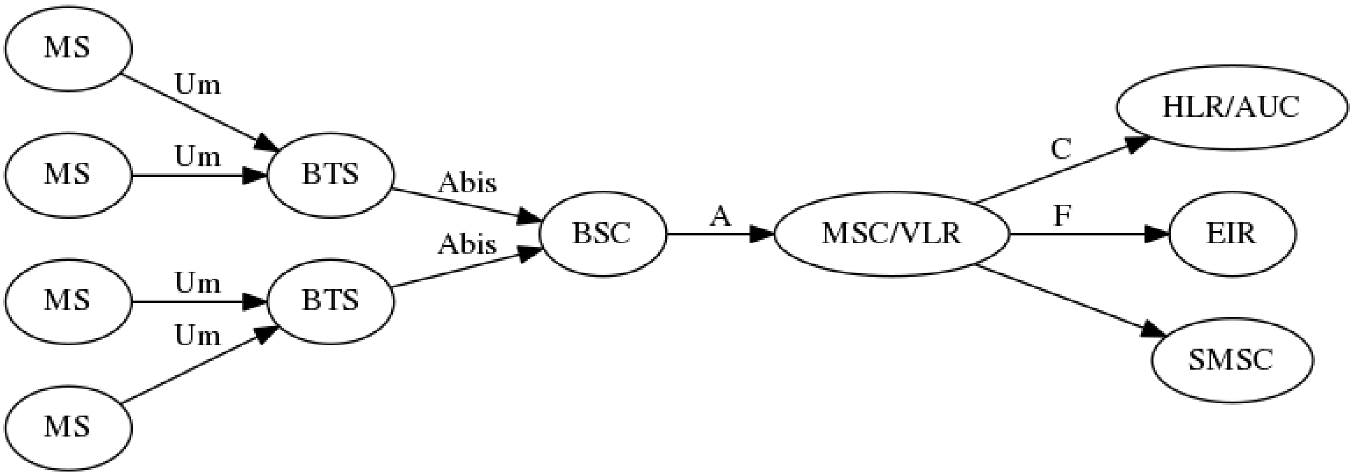

Figure 1: Classic GSM network architecture (simplified)

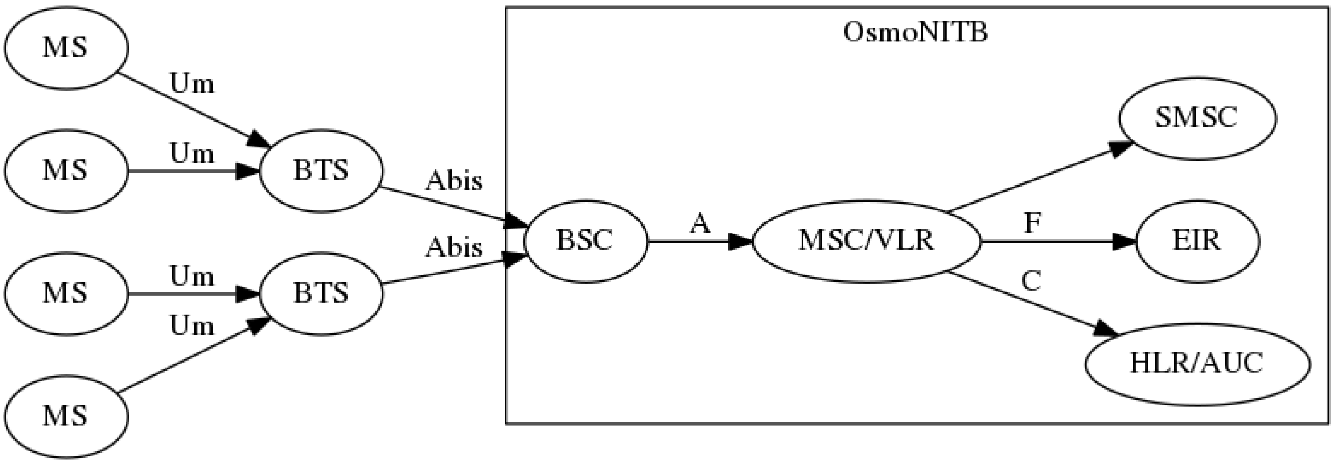

Figure 2: GSM system architecture using OsmoNITB

4.2 Software Components

OsmoNITB contains a variety of different software components, which we'll quickly describe in this section.

4.2.1 A-bis Implementation

OsmoNITB implements the ETSI/3GPP specified A-bis interface, including *3GPP TS 48.056* [3gpp-ts-48-056] (LAPD), *3GPP TS 48.058* [3gpp-ts-48-058] (RSL) and 3GPP TS 52.021 [3gpp-ts-52-021] (OML). In addition, it supports a variety of vendor-specific extensions and dialects in order to communicate with BTSs from Siemens, Nokia, Ericsson, ip.access and sysmocom.

For more information, see Section 11 and Section 12.

4.2.2 BSC Implementation

The BSC implementation covers the classic functionality of a GSM Base Station Controller, i.e.

- configuring and bringing up BTSs with their TRXs and TSs
- implementing the A-bis interface / protocols for signalling and actual voice data (TRAU frames).
- processing measurement results from the mobile stations in dedicated mode, performing hand-over decision and execution.
- Terminating the *3GPP TS 24.008* [3gpp-ts-24-008] RR (Radio Resource) sub-layer from the MS.

For more information, see Section 9, Section 11 and Section 12.

4.2.3 HLR/AUC

A minimalistic implementation of the subscriber database (HLR) and subscriber secret key storage (AUC).

For more information, see Section 13.

4.2.4 SMSC

A minimal store-and-forward server for SMS, supporting both MO and MT SMS service, as well as multi-part messages.

The built-in SMSC also supports an external SMSC interface. For more information, see Section 14.

4.2.5 MSC

The MSC component of OsmoNITB implements the mobility management (MM) functions of the TS 04.08, as well as the optional security related procedures for cryptographic authentication and encryption.

Furthermore, it can handle TS 04.08 Call Control (CC), either by use of an internal MNCC handler, or by use of an external MNCC agent. For more information see Section 15.

4.2.6 TRAU mapper / E1 sub-channel muxer

Unlike classic GSM networks, OsmoNITB does not perform any transcoding. Rather, a compatible codec is selected for both legs of a call, and codec frames are passed through transparently. In order to achieve this with E1 based BTS, OsmoNITB contains a E1 sub-channel de- and re-multiplexer as well as a TRAU mapper that can map uplink to downlink frames and vice versa.

4.2.7 RTP proxy

BTS models implementing A-bis over IP don't use classic TRAU frames but typically transport the voice codec frames as RTP/UDP/IP protocol. OsmoNITB can either instruct the BTSs to send those voice streams directly to each other (BTS to BTS without any intermediary), or it can run an internal RTP proxy for passing frames from one BTS to another.

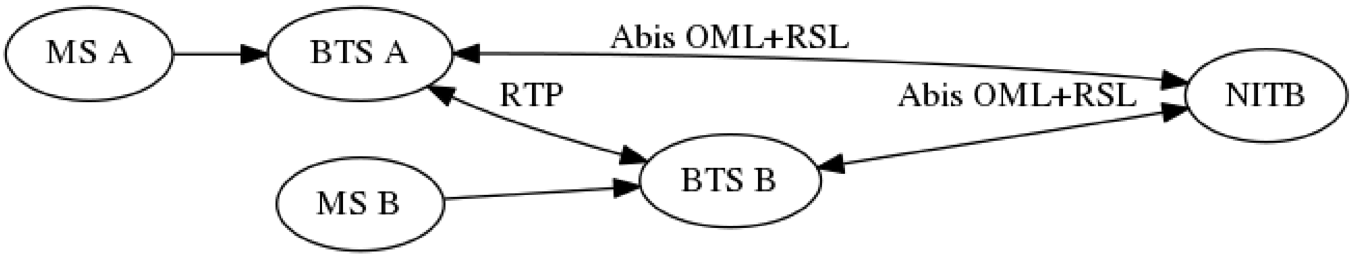

Figure 3: RTP flow without RTP proxy mode (default)

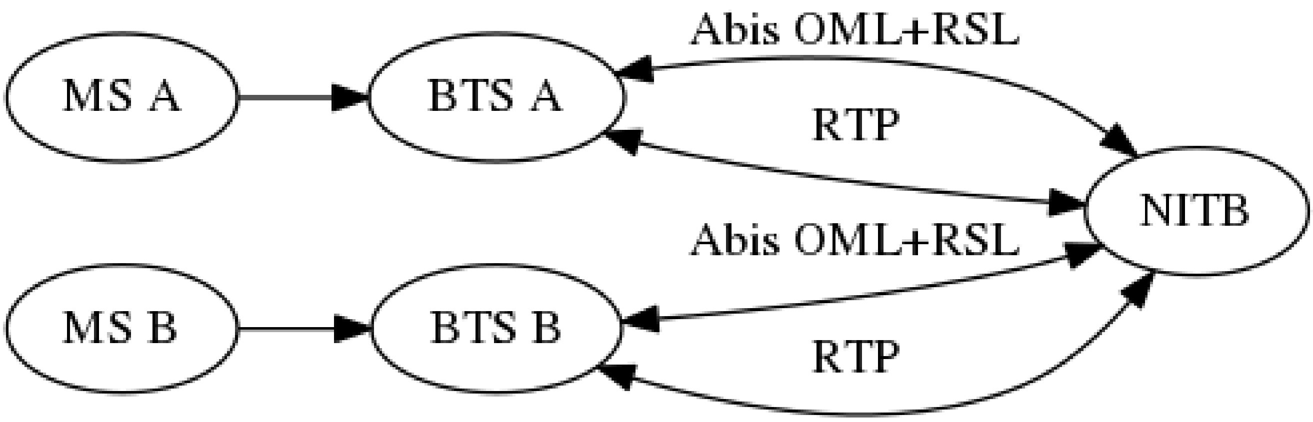

Figure 4: RTP flow with RTP proxy mode

5 Running OsmoNITB

The OsmoNITB executable (`osmo-nitb`) offers the following command-line arguments:

5.1 SYNOPSIS

osmo-nitb [-h|-V] [-d *DBGMASK*] [-D] [-c *CONFIGFILE*] [-s] [-T] [-e *LOGLEVEL*] [-l *DATABASE*] [-a] [-P] [-m] [-C] [-r *RFCTL*]

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, --debug *DBGMASK,DBGLEVELS*
Set the log subsystems and levels for logging to stderr. This has mostly been superseded by VTY-based logging configuration, see Section 8 for further information.

-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 `openbsc.cfg` in the current working directory.

-s, --disable-color
Disable colors for logging to stderr. This has mostly been deprecated by VTY based logging configuration, see Section 8 for more information.

-T, --timestamp
Enable time-stamping of log messages to stderr. This has mostly been deprecated by VTY based logging configuration, see Section 8 for more information.

-e, --log-level ***LOGLEVEL***
Set the global log level for logging to stderr. This has mostly been deprecated by VTY based logging configuration, see Section 8 for more information.

-l, --database ***DATABASE***
Specify the file name of the SQLite3 database to use as HLR/AUC storage

-a, --authorize-everyone
Authorize every subscriber to the network. This corresponds to the `auth-policy open` VTY configuration option.

WARNING
This is dangerous as you may disrupt services to subscribers that are not part of your network! Don't use unless you absolutely know what you're doing!

-P, --rtp-proxy
Enable the RTP proxy code inside OsmoNITB. This will force all voice RTP data to pass through OsmoNITB, rather than going directly from BTS to MGW, or BTS to BTS.

-M, --mncc-sock-path
Enable the MNCC socket for an external MNCC handler. See Section 15 for further information.

-m, --mncc-sock
Same as option -M (deprecated).

-C, --no-dbcounter
Disable the regular periodic synchronization of statistics counters to the database.

-r, --rf-ctl ***RFCTL***
Offer a Unix domain socket for RF control at the path/filename *RFCTL* in the file system.

5.3 Multiple instances

Running multiple instances of `osmo-nitb` is possible if all 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
```

The OML interface also needs to be separated by binding it to different IP addresses. Usually it is not possible to use addresses from the loopback address range here since the OML interface needs to be reachable by an external BTS. If only one ethernet interface is available, sub-devices with different IP addresses can be created.

Example: Binding OML to a specific IP address

```
e1_input
 ipa bind 10.9.1.101
```

Note
Depending on the application, it is necessary to have different ARFCN, MCC, MNC and network name settings. It might also be necessary to point to different database and config files using command line options (see option -l and -c).

> **Note**
> If an external MNCC handler is used, the user has to assign a different socket path to reach osmo-nitb instance using commandline option -M. If option -M is left out, the internal MNCC handler is used and no further configuration is required

6 Control interface

The actual protocol is described in Section 16, the variables common to all programs using it are described in Section 16.2. The variables shared with OsmoBSC are described in corresponding section of OsmoBSC documentation.Here we describe variables specific to OsmoNITB.

Table 1: Variables available over control interface

Name	Access	Trap	Value	Comment
subscriber-modify-v1	WO	No	"<imsi>,<msisdn>,<alg>,<ki>"	See Section 6.1 for details.
subscriber-delete-v1	WO	No	"<imsi>"	See Section 6.2 for details.
subscriber-list-active-v1	RO	No		Return list of active subscribers.

6.1 subscriber-modify-v1

Modify (or add if missing) subscriber entry with the give IMSI, MSISDN, Ki and algorithm (valid values are "none", "xor" and "comp128v1"). The subscriber is automatically marked as authorized.

6.2 subscriber-delete-v1

Delete the subscriber with the given IMSI. Returns "Removed active subscriber" or "Removed" depending on the subscriber's use status.

The following variables are only available over control interface of osmo-bsc_nat program.

Table 2: Variables available over control interface of osmo-bsc_nat

Name	Access	Trap	Value	Comment
net.0.bsc.N.*	RW	Yes	Arbitrary variable	Forward given command to BSC N control interface.
net.0.bsc_cfg.N.access-list-name	RW	No	"<name>"	Set/Get ACL for a given BSC N.
net.0.bsc_cfg.N.no-access-list-name	WO	No	Ignored	Remove ACL for a given BSC N.
net.0.add.allow.access-list.A	WO	No	"<regexp>"	See Section 6.3 for details.
net.0.save-configuration	WO	No	Ignored	Save current running config into file.
net.0.bsc.N.notification-rejection-v1	NA	Yes	"imsi=<imis>"	See Section 6.4 for details.

6.3 allow.access-list

Add given regular expression for matching IMSI(s) to allowed access list A.

6.4 notification-rejection-v1

This TRAP event notifies all connected clients about IMSI which was rejected by BSC N.

7 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 3: 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

7.1 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.

7.2 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.

7.3 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.

7.3.1 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".

7.3.2 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.

7.3.3 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>
```

8 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.

8.1 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 ?`

8.2 Log levels

For each of the log categories (see Section 8.1), 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`.

8.3 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`

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

8.4 Log targets

Each of the log targets represent certain destination for log messages. It can be configured independently by selecting levels (see Section 8.2) for categories (see Section 8.1) as well as filtering (see Section 8.3) and other options like `logging timestamp` for example.

8.4.1 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 8.1 for more details on categories and Section 8.2 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 8.3.

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.

8.4.2 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.

8.4.3 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.

8.4.4 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.

8.4.5 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)#
```

9 OsmoNITB Core Network Subsystem

The OsmoNITB Core Network is a minimalistic implementation of the classic MSC/VLR/HLR/AUC/SMSC components. None of the standardized core network protocols (such as SCCP/TCAP/MAP) are used, interfaces between VLR and HLR are simple function calls inside the same software package.

OsmoNITB can thus provide autonomous voice and SMS services to its coverage area, but it cannot provide roaming interfaces to classic GSM operators. To support this configuration, it is suggested to use the OsmoBSC variant of OpenBSC and interface it with a conventional MSC using A-over-IP protocol.

If you have classic GSM network/operator background, many of the concepts used in OsmoNITB will appear foreign to you, as they are very unlike the conventional GSM networks that you have worked with.

9.1 Configuring the Core Network

Like everything else, the core network related parameters are configured using the VTY. The respective parameters are underneath the `network` config node.

You can get to that node by issuing the following commands:

Entering the config network node

```
OpenBSC> enable
OpenBSC# configure terminal
OpenBSC(config)# network
OpenBSC(config-net)#
```

A full reference to them can be found in the *OsmoNITB VTY reference manual* [vty-ref-osmonitb]. This section will only introduce the most commonly used settings in detail.

Tip
You can always use the `list` VTY command to get a list of all possible commands at the current node.

9.2 Configuring the MCC/MNC

The key identities of every GSM PLMN is the MCC and MNC. They are identical over the entire network. In most cases, the MCC/MNC will be allocated to the operator by the respective local regulatory authority. For example, to set the MCC/MNC of 262-89, you may enter:

Configuring the MCC/MNC

```
OpenBSC(config-net)# network country code 262
OpenBSC(config-net)# mobile network code 89
```

9.3 Configuring MM INFO

The *MM INFO* procedure can be used after a successful *LOCATION UPDATE* in order to transmit the human-readable network name as well as local time zone information to the MS.

By default, MM INFO is not active. You can activate it, and set its configuration using the VTY. An example is provided below.

Configuring MM INFO

```
OpenBSC(config-net)# mm info 1
OpenBSC(config-net)# short name OpenBSC
OpenBSC(config-net)# long name OpenBSC
```

Note
Not all phone support the MM INFO procedure. Unless they already are factory-programmed to contain the name for your MCC/MNC, then they will likely only provide a numeric display of the network name, such as *262-89* or with the country code transformed into a letter, such as *D 89*.

The time information transmitted is determined by the local system time of the operating system on which OsmoNITB is running. As BTSs attached to one OsmoNITB can reside in different time zones, it is possible to use the `timezone` command at each BTS node to set different time zone offsets in hours and quarter hours.

9.4 Setting the NECI bit

NECI (New Establishment Cause Indication) is an optional change of the definition for establishment cause in the RACH burst. Among other things, in a network with NECI, a MS can explicitly indicate its TCH/H capability while asking for a dedicated radio channel.

It is strongly recommended to use NECI. You can do so by issuing the following command: .Enabling NECI

```
OpenBSC(config-net)# neci 1
```

9.5 Configuring Handover

As opposed to cell re-selection in idle mode, handover refers to the explicit transfer of a MS dedicated channel from one radio channel to another. This typically happens due to a MS moving from one cell to another while in an active call.

OsmoNITB has a number of hand-over related parameters by which the hand-over algorithm can be tuned. Logically, those settings are settings of the BSC component, but for historic reasons, they are also configured under the *network* VTY node.

Configuring Handover

```
OpenBSC(config-net)# handover 1
OpenBSC(config-net)# handover window rxlev averaging 10
OpenBSC(config-net)# handover window rxqual averaging 1
OpenBSC(config-net)# handover window rxlev neighbor averaging 10
OpenBSC(config-net)# handover power budget interval 6
OpenBSC(config-net)# handover power budget hysteresis 3
OpenBSC(config-net)# handover maximum distance 9999
```

Note
If you are receiving the following error message:

```
OpenBSC(config-net)# handover 1
% Cannot enable handover unless RTP Proxy mode is enabled by using the -P command line ↩
    option
```

then you should do as indicated and make sure to start your `osmo-nitb` process using the `-P` command line option.

10 BSC level configuration

The BSC component is shared between OsmoBSC and OsmoNITB. This chapter describes some of the configuration options related to this shared BSC component.

10.1 Hand-over

10.1.1 Hand-over in GSM

Hand-over is the process of changing a MS with a currently active dedicated channel from one BTS to another BTS. As opposed to idle mode, where the MS autonomously performs cell re-selection, in dedicated mode this happens under network control.

In order to determine when to perform hand-over, and to which cells, the network requests the MS to perform measurements on a list of neighbor cell channels, which the MS then reports back to the network in the form of GSM RR *Measurement Result* messages. Those messages contain the downlink measurements as determined by the MS.

Furthermore, the BTS also performs measurements on the uplink, and communicates those by means of RSL to the BSC.

The hand-over decision is made by an algorithm that processes those measurement results and determines when to perform the hand-over.

10.1.2 Configuration of hand-over in OsmoBSC/OsmoNITB

OsmoBSC (like the internal BSC component of OsmoNITB) only support so-called intra-BSC hand-over, where the hand-over is performed between two BTSs within the same BSC.

Hand-over is enabled and configured by the use of a set of `handover` commands. Using those, you can tune the key parameters of the hand-over algorithm and adapt it to your specific environment.

Example handover configuration snippet

```
handover 1 ❶
handover window rxlev averaging 10 ❷
handover window rxqual averaging 1 ❸
handover window rxlev neighbor averaging 10 ❹
handover power budget interval 6 ❺
handover power budget hysteresis 3 ❻
handover maximum distance 9999 ❼
```

❶ Enable hand-over

❷ Set the RxLev averaging window for the serving cell to 10 measurements

❸ Set the RxQual averaging window for the serving cell to 1 measurement (no window)

❹ Set the RxLev averaging for neighbor cells to 10 measurements

❺ Check for the conditions of a power budget hand-over every 6 SACCH frames

❻ A neighbor cell must be at least 3 dB stronger than the serving cell to be considered a candidate for hand-over

❼ Perform a maximum distance hand-over if TA is larger 9999 (i.e. never)

10.2 Timer Configuration

The GSM specification specifies a variety of timers both on the network as well as on the mobile station side.

Those timers can be configured using the `timer tXXXX` command.

Table 4: Configurable Timers

Table 4: (continued)

node	timer	default	description
network	t3101	10	Timeout for *Immediate Assignment* (sec)
network	t3103	?	Timeout for Handover (sec)
network	t3105	40	Repetition of *Physical Information* (sec)
network	t3107	?	?
network	t3109	?	RSL SACCH deactivation timeout (sec)
network	t3111	?	RSL timeout to wait before releasing the RF channel (sec)
network	t3113	60	Time to try paging for a subscriber (sec)
network	t3115	?	?
network	t3117	?	?
network	t3119	?	?
network	t3122	10	Waiting time after *Immediate Assignment Reject*
network	t3141	?	?

10.3 Discontinuous Transmission (DTX)

GSM provides a full-duplex voice call service. However, in any civilized communication between human beings, only one of the participants is speaking at any given point in time. This means that most of the time, one of the two directions of the radio link is transmitting so-called *silence frames*.

During such periods of quiescence in one of the two directions, it is possible to suppress transmission of most of the radio bursts, as there is no voice signal to transport. GSM calls this feature *Discontinuous Transmission*. It exists separately for uplink (DTXu) and downlink (DTXd).

Downlink DTX is only permitted on non-primary transceivers (!= TRX0), as TRX0 must always transmit at constant output power to ensure it is detected during cell selection.

Uplink DTX is possible on any TRX, and serves primarily two uses:

possible on any TRX, and serves primarily two uses:

1. reducing the MS battery consumption by transmitting at a lower duty cycle
2. reducing the uplink interference caused in surrounding cells that re-use the same ARFCN.

DTS for both uplink and downlink is implemented in the BTS. Not all BTS models support it.

The Osmocom BSC component can instruct the BTS to enable or disable uplink and/or downlink DTX by means of A-bis OML.

11 Reviewing and Provisioning BTS configuration

The main functionality of the BSC component is to manage BTSs. As such, provisioning BTSs within the BSC is one of the most common tasks during BSC operation. Just like about anything else in OpenBSC, they are configured using the VTY.

BTSs are internally numbered with integer numbers starting from "0" for the first BTS. BTS numbers have to be contiguous, so you cannot configure 0,1,2 and then 5.

11.1 Reviewing current BTS status and configuration

In order to view the status and properties of a BTS, you can issue the `show bts` command. If used without any BTS number, it will display information about all provisioned BTS numbers.

```
OpenBSC> show bts 0
BTS 0 is of nanobts type in band DCS1800, has CI 0 LAC 1, BSIC 63, TSC 7 and 1 TRX
Description: (null)
MS Max power: 15 dBm
Minimum Rx Level for Access: -110 dBm
Cell Reselection Hysteresis: 4 dBm
RACH TX-Integer: 9
RACH Max transmissions: 7
System Information present: 0x0000007e, static: 0x00000000
  Unit ID: 200/0/0, OML Stream ID 0xff
  NM State: Oper 'Enabled', Admin 2, Avail 'OK'
  Site Mgr NM State: Oper 'Enabled', Admin 0, Avail 'OK'
  Paging: 0 pending requests, 0 free slots
  OML Link state: connected.
  Current Channel Load:
                   TCH/F:   0% (0/5)
                  SDCCH8:   0% (0/8)
```

You can also review the status of the TRXs configured within the BTSs of this BSC by using `show trx`:

```
OpenBSC> show trx 0 0
TRX 0 of BTS 0 is on ARFCN 871
Description: (null)
  RF Nominal Power: 23 dBm, reduced by 0 dB, resulting BS power: 23 dBm
  NM State: Oper 'Enabled', Admin 2, Avail 'OK'
  Baseband Transceiver NM State: Oper 'Enabled', Admin 2, Avail 'OK'
  ip.access stream ID: 0x00
```

The output can be restricted to the TRXs of one specified BTS number (`show trx 0`) or even that of a single specified TRX within a specified BTS (`show trx 0 0`).

Furthermore, information on the individual timeslots can be shown by means of `show timeslot`. The output can be restricted to the timeslots of a single BTS (`show timeslot 0`) or that of a single TRX (`show timeslot 0 0`). Finally, you can restrict the output to a single timeslot by specifying the BTS, TRX and TS numbers (`show timeslot 0 0 4`).

```
OpenBSC> show timeslot 0 0 0
BTS 0, TRX 0, Timeslot 0, phys cfg CCCH, TSC 7
  NM State: Oper 'Enabled', Admin 2, Avail 'OK'
OpenBSC> show timeslot 0 0 1
BTS 0, TRX 0, Timeslot 1, phys cfg SDCCH8, TSC 7
  NM State: Oper 'Enabled', Admin 2, Avail 'OK'
```

11.2 Provisioning a new BTS

In order to provision BTSs, you have to enter the BTS config node of the VTY. In order to configure BTS 0, you can issue the following sequence of commands:

```
OpenBSC> enable
OpenBSC# configure terminal
OpenBSC(config)# network
OpenBSC(config-net)# bts 0
OpenBSC(config-net-bts)#
```

At this point, you have a plethora of commands, in fact an entire hierarchy of commands to configure all aspects of the BTS, as well as each of its TRX and each timeslot within each TRX. For a full reference, please consult the respective chapter in the VTY reference of OpenBSC.

BTS configuration depends quite a bit on the specific BTS vendor and model. The section below provides just one possible example for the case of a sysmoBTS.

```
OpenBSC(config-net-bts)# type sysmobts
OpenBSC(config-net-bts)# band DCS1800
OpenBSC(config-net-bts)# description The new BTS in Baikonur
OpenBSC(config-net-bts)# location_area_code 2342
OpenBSC(config-net-bts)# cell_identity 5
OpenBSC(config-net-bts)# base_station_id_code 63
OpenBSC(config-net-bts)# ip.access unit_id 8888 0
OpenBSC(config-net-bts)# ms max power 40
OpenBSC(config-net-bts)# trx 0
OpenBSC(config-net-bts-trx)# arfcn 871
OpenBSC(config-net-bts-trx)# nominal power 23
OpenBSC(config-net-bts-trx)# max_power_red 0
OpenBSC(config-net-bts-trx)# timeslot 0
OpenBSC(config-net-bts-trx-ts)# phys_chan_config CCCH+SDCCH4
OpenBSC(config-net-bts-trx-ts)# exit
OpenBSC(config-net-bts-trx)# timeslot 1
OpenBSC(config-net-bts-trx-ts)# phys_chan_config TCH/F
OpenBSC(config-net-bts-trx-ts)# exit
OpenBSC(config-net-bts-trx)# timeslot 2
OpenBSC(config-net-bts-trx-ts)# phys_chan_config TCH/F
OpenBSC(config-net-bts-trx-ts)# exit
OpenBSC(config-net-bts-trx)# timeslot 3
OpenBSC(config-net-bts-trx-ts)# phys_chan_config TCH/F
OpenBSC(config-net-bts-trx-ts)# exit
OpenBSC(config-net-bts-trx)# timeslot 4
OpenBSC(config-net-bts-trx-ts)# phys_chan_config TCH/F
OpenBSC(config-net-bts-trx-ts)# exit
OpenBSC(config-net-bts-trx)# timeslot 5
OpenBSC(config-net-bts-trx-ts)# phys_chan_config TCH/F
OpenBSC(config-net-bts-trx-ts)# exit
OpenBSC(config-net-bts-trx)# timeslot 6
OpenBSC(config-net-bts-trx-ts)# phys_chan_config TCH/F
OpenBSC(config-net-bts-trx-ts)# exit
OpenBSC(config-net-bts-trx)# timeslot 7
OpenBSC(config-net-bts-trx-ts)# phys_chan_config PDCH
OpenBSC(config-net-bts-trx-ts)# exit
```

11.3 System Information configuration

A GSM BTS periodically transmits a series of *SYSTEM INFORMATION* messages to mobile stations, both via the BCCH in idle mode, was well as via the SACCH in dedicated mode. There are many different types of such messages. For their detailed contents and encoding, please see *3GPP TS 24.008* [3gpp-ts-24-008].

For each of the *SYSTEM INFORMATION* message types, you can configure to have the BSC generate it automatically (*computed*), or you can specify the respective binary message as a string of hexadecimal digits.

The default configuration is to compute all (required) *SYSTEM INFORMATION* messages automatically.

Please see the *OsmoBSC VTY Reference Manual* [vty-ref-osmobsc] for further information, particularly on the following commands:

- `system-information (1|2|3|4|5|6|7|8|9|10|13|16|17|18|19|20|2bis|2ter|2quater|5bis|5ter) mode (static|computed)`
- `system-information (1|2|3|4|5|6|7|8|9|10|13|16|17|18|19|20|2bis|2ter|2quater|5bis|5ter) static HEXSTRING`

11.4 Neighbor List configuration

Every BTS sends a list of ARFCNs of neighbor cells . within its *SYSTEM INFORMATION 2* (and 2bis/2ter) messages on the BCCH . within its *SYSTEM INFORMATION 5* messages on SACCH in dedicated mode

For every BTS config node in the VTY, you can specify the behavior of the neighbor list using the `neighbor list mode` VTY command:

automatic
: Automatically generate a list of neighbor cells using all other BTSs configured in the VTY

manual
: Manually specify the neighbor list by means of `neighbor-list (add|del) arfcn <0-1023>` commands, having identical neighbor lists on BCCH (SI2) and SACCH (SI5)

manual-si5
: Manually specify the neighbor list by means of `neighbor-list (add|del) arfcn <0-1023>` for BCCH (SI2) and a separate neighbor list by means of `si5 neighbor-list (add|del) arfcn <0-1023>` for SACCH (SI5).

11.5 Configuring GPRS PCU parameters of a BTS

In the case of BTS models using Abis/IP (IPA), the GPRS PCU is located inside the BTS. The BTS then establishes a Gb connection to the SGSN.

All the BTS-internal PCU configuration is performed via A-bis OML by means of configuring the *CELL*, *NSVC* (NS Virtual Connection and *NSE* (NS Entity).

There is one *CELL* node and one *NSE* node, but there are two *NSVC* nodes. At the time of this writing, only the NSVC 0 is supported by OsmoBTS, while both NSVC are supported by the ip.access nanoBTS.

The respective VTY configuration parameters are described below. They all exist beneath each BTS VTY config node.

But let's first start with a small example

Example configuration of GPRS PCU parameters at VTY BTS node

```
OpenBSC(config-net-bts)# gprs mode gprs
OpenBSC(config-net-bts)# gprs routing area 1
OpenBSC(config-net-bts)# gprs cell bvci 1234
OpenBSC(config-net-bts)# gprs nsei 1234
OpenBSC(config-net-bts)# gprs nsvc 0 nsvci 1234
OpenBSC(config-net-bts)# gprs nsvc 0 local udp port 23000
OpenBSC(config-net-bts)# gprs nsvc 0 remote udp port 23000
OpenBSC(config-net-bts)# gprs nsvc 0 remote ip 192.168.100.239
```

11.6 More explanation about the PCU config parameters

11.6.1 `gprs mode (none|gprs|egprs)`

This command determines if GPRS (or EGPRS) services are to be enabled in this cell at all.

11.6.2 `gprs cell bvci <2-65535>`

Configures the *BSSGP Virtual Circuit Identifier*. It must be unique between all BGSGP connections to one SGSN.

Note
It is up to the system administrator to ensure all PCUs are allocated an unique bvci. OsmoBSC will not ensure this policy.

11.6.3 gprs nsei <0-65535>

Configures the *NS Entity Identifier*. It must be unique between all NS connections to one SGSN.

Note
It is up to the system administrator to ensure all PCUs are allocated an unique bvci. OsmoBSC will not ensure this policy.

11.6.4 gprs nsvc <0-1> nsvci <0-65535>

Configures the *NS Virtual Connection Identifier*. It must be unique between all NS virtual connections to one SGSN.

Note
It is up to the system administrator to ensure all PCUs are allocated an unique nsvci. OsmoBSC will not ensure this policy.

11.6.5 gprs nsvc <0-1> local udp port <0-65535>

Configures the local (PCU side) UDP port for the NS-over-UDP link.

11.6.6 gprs nsvc <0-1> remote udp port <0-65535>

Configures the remote (SGSN side) UDP port for the NS-over-UDP link.

11.6.7 gprs nsvc <0-1> remote ip A.B.C.D

Configures the remote (SGSN side) UDP port for the NS-over-UDP link.

11.6.8 gprs ns timer (tns-block|tns-block-retries|tns-reset|tns-reset-retries|tns-test|tns-alive|tns-alive-retries) <0-255>

Configures the various GPRS NS related timers. Please check the GPRS NS specification for the detailed meaning of those timers.

11.7 Dynamic Timeslot Configuration (TCH / PDCH)

A dynamic timeslot is in principle a voice timeslot (TCH) that is used to serve GPRS data (PDCH) when no voice call is active on it. This enhances GPRS bandwidth while no voice calls are active, which is dynamically scaled down as voice calls need to be served. This is a tremendous improvement in service over statically assigning a fixed number of timeslots for voice and data.

Dynamic timeslots work both with OsmoNITB as well as with OsmoBSC driven by a third-party MSC. The causality is as follows: to establish a voice call, the MSC requests a logical channel of a given TCH kind from the BSC. The BSC assigns such a channel from a BTS' TRX's timeslot of its choice. The knowledge that a given timeslot is dynamic exists only on the BSC level. When the MSC asks for a logical channel, the BSC may switch off PDCH on a dynamic timeslot and then assign a logical TCH channel on it. Hence, though compatibility with the BTS needs to be ensured, any MSC is compatible with dynamic timeslots by definition.

OsmoBSC and OsmoNITB support two kinds of dynamic timeslot handling, configured via the `network` / `bts` / `trx` / `timeslot` / `phys_chan_config` configuration. Not all BTS models support dynamic channels.

Table 5: Dynamic timeslot support by various BTS models

Table 5: (continued)

	TCH/F_TCH/H_PDCH	TCH/F_PDCH
ip.access nanoBTS	-	supported
Ericsson RBS	supported	-
sysmoBTS using *osmo-bts-sysmo*	supported	supported
various SDR platforms using *osmo-bts-trx*	supported	supported
Nutaq Litecell 1.5 using *osmo-bts-litecell15*	supported	supported
Octasic OctBTS using *osmo-bts-octphy*	-	-

The *OsmoBTS Abis Protocol Specification* [osmobts-abis-spec] describes the non-standard RSL messages used for these timeslot kinds.

Note
Same as for dedicated PDCH timeslots, you need to enable GPRS and operate a PCU, SGSN and GGSN to provide the actual data service.

11.7.1 Osmocom Style Dynamic Timeslots (TCH/F_TCH/H_PDCH)

Timeslots of the `TCH/F_TCH/H_PDCH` type dynamically switch between TCH/F, TCH/H and PDCH, depending on the channel kind requested by the MSC. The RSL messaging for `TCH/F_TCH/H_PDCH` timeslots is compatible with Ericsson RBS.

BTS models supporting this timeslot kind are shown in Table 5.

Note
At the time of writing, OsmoNITB disables TCH/F on this timeslot type due to transcoding limitations. Operation of OsmoBSC with a third-party MSC is not affected by this limitation. See https://osmocom.org/issues/1778.

11.7.2 ip.access Style Dynamic Timeslots (TCH/F_PDCH)

Timeslots of the `TCH/F_PDCH` type dynamically switch between TCH/F and PDCH. The RSL messaging for `TCH/F_PDCH` timeslots is compatible with ip.access nanoBTS.

BTS models supporting this timeslot kind are shown in Table 5.

11.7.3 Avoid PDCH Exhaustion

To avoid disrupting GPRS, configure at least one timeslot as dedicated PDCH. With only dynamic timeslots, a given number of voice calls would convert all timeslots to TCH, and no PDCH timeslots would be left for GPRS service.

11.7.4 Dynamic Timeslot Configuration Examples

This is an extract of an *osmo-nitb* or *openbsc* config file. A timeslot configuration with five Osmocom style dynamic timeslots and one dedicated PDCH may look like this:

```
network
 bts 0
  trx 0
   timeslot 0
    phys_chan_config CCCH+SDCCH4
   timeslot 1
    phys_chan_config SDCCH8
```

```
   timeslot 2
    phys_chan_config TCH/F_TCH/H_PDCH
   timeslot 3
    phys_chan_config TCH/F_TCH/H_PDCH
   timeslot 4
    phys_chan_config TCH/F_TCH/H_PDCH
   timeslot 5
    phys_chan_config TCH/F_TCH/H_PDCH
   timeslot 6
    phys_chan_config TCH/F_TCH/H_PDCH
   timeslot 7
    phys_chan_config PDCH
```

With the ip.access nanoBTS, only `TCH/F_PDCH` dynamic timeslots are supported, and hence a nanoBTS configuration may look like this:

```
network
 bts 0
  trx 0
   timeslot 0
    phys_chan_config CCCH+SDCCH4
   timeslot 1
    phys_chan_config SDCCH8
   timeslot 2
    phys_chan_config TCH/F_PDCH
   timeslot 3
    phys_chan_config TCH/F_PDCH
   timeslot 4
    phys_chan_config TCH/F_PDCH
   timeslot 5
    phys_chan_config TCH/F_PDCH
   timeslot 6
    phys_chan_config TCH/F_PDCH
   timeslot 7
    phys_chan_config PDCH
```

12 OsmoNITB example configuration files

The `openbsc/doc/examples/osmo-nitb` directory in the OpenBSC source tree contains a collection of example configuration files, sorted by BTS type.

This chapter is illustrating some excerpts from those examples

12.1 Example configuration for OsmoNITB with one dual-TRX BS-11

Example 12.1 OsmoNITB with BS11, 2 TRX, no frequency hopping

```
e1_input
 e1_line 0 driver misdn
network
 network country code 1
 mobile network code 1
 short name OpenBSC
 long name OpenBSC
 timer t3101 10
 timer t3113 60
 bts 0
  type bs11 ❶
```

```
  band GSM900
  cell_identity 1
  location_area_code 1
  training_sequence_code 7
  base_station_id_code 63
  oml e1 line 0 timeslot 1 sub-slot full ❷
  oml e1 tei 25 ❸
  trx 0
   arfcn 121
   max_power_red 0
   rsl e1 line 0 timeslot 1 sub-slot full ❹
   rsl e1 tei 1 ❺
    timeslot 0
     phys_chan_config CCCH+SDCCH4
     e1 line 0 timeslot 1 sub-slot full
    timeslot 1
     phys_chan_config TCH/F
     e1 line 0 timeslot 2 sub-slot 1 ❻
    timeslot 2
     phys_chan_config TCH/F
     e1 line 0 timeslot 2 sub-slot 2
    timeslot 3
     phys_chan_config TCH/F
     e1 line 0 timeslot 2 sub-slot 3
    timeslot 4
     phys_chan_config TCH/F
     e1 line 0 timeslot 3 sub-slot 0
    timeslot 5
     phys_chan_config TCH/F
     e1 line 0 timeslot 3 sub-slot 1
    timeslot 6
     phys_chan_config TCH/F
     e1 line 0 timeslot 3 sub-slot 2
    timeslot 7
     phys_chan_config TCH/F
     e1 line 0 timeslot 3 sub-slot 3
  trx 1
   arfcn 123
   max_power_red 0
   rsl e1 line 0 timeslot 1 sub-slot full ❼
   rsl e1 tei 2 ❽
    timeslot 0
     phys_chan_config TCH/F
     e1 line 0 timeslot 4 sub-slot 0 ❾
    timeslot 1
     phys_chan_config TCH/F
     e1 line 0 timeslot 4 sub-slot 1
    timeslot 2
     phys_chan_config TCH/F
     e1 line 0 timeslot 4 sub-slot 2
    timeslot 3
     phys_chan_config TCH/F
     e1 line 0 timeslot 4 sub-slot 3
    timeslot 4
     phys_chan_config TCH/F
     e1 line 0 timeslot 5 sub-slot 0
    timeslot 5
     phys_chan_config TCH/F
     e1 line 0 timeslot 5 sub-slot 1
    timeslot 6
     phys_chan_config TCH/F
     e1 line 0 timeslot 5 sub-slot 2
```

```
    timeslot 7
     phys_chan_config TCH/F
     e1 line 0 timeslot 5 sub-slot 3
```

1. The BTS type must be set to *bs11*
2. The OML E1 timeslot needs to be identical with what was on the BTS side using LMT.
3. The OML TEI value needs to be identical with what was configured on the BTS side using LMT.

4, 7 The RSL E1 timeslot can be identical for all TRX.

5, 8 The RSL TEI values *must* be different if multiple TRX share one E1 signalling timeslot.

6, 9 The TCH all need to be allocated one 16k sub-slot on the E1

12.2 Example configuration for OsmoNITB with one single-TRX nanoBTS

Example 12.2 OsmoNITB with one single-TRX nanoBTS

```
e1_input
 e1_line 0 driver ipa ❶
network
 network country code 1
 mobile network code 1
 short name OpenBSC
 long name OpenBSC
 auth policy closed
 location updating reject cause 13
 encryption a5 0
 neci 1
 rrlp mode none
 mm info 1
 handover 0
 bts 0
  type nanobts ❷
  band DCS1800 ❸
  cell_identity 0
  location_area_code 1
  training_sequence_code 7
  base_station_id_code 63
  ms max power 15
  cell reselection hysteresis 4
  rxlev access min 0
  channel allocator ascending
  rach tx integer 9
  rach max transmission 7
  ip.access unit_id 1801 0 ❹
  oml ip.access stream_id 255 line 0
  gprs mode none
  trx 0
   rf_locked 0
   arfcn 871 ❺
   nominal power 23
   max_power_red 20 ❻
   rsl e1 tei 0
    timeslot 0
     phys_chan_config CCCH+SDCCH4
    timeslot 1
     phys_chan_config SDCCH8
```

```
    timeslot 2
     phys_chan_config TCH/F
    timeslot 3
     phys_chan_config TCH/F
    timeslot 4
     phys_chan_config TCH/F
    timeslot 5
     phys_chan_config TCH/F
    timeslot 6
     phys_chan_config TCH/F
    timeslot 7
     phys_chan_config TCH/F
```

1. You have to configure one virtual E1 line with the IPA driver in order to use Abis/IP. One e1_line is sufficient for any number of A-bis/IP BTSs, there is no limit like in physical E1 lines.
2. The BTS type must be set using `type nanobts`
3. The GSM band must be set according to the BTS hardware.
4. The IPA Unit ID parameter must be set to what has been configured on the BTS side using the *BTS Manager* or `ipaccess-config`.
5. The ARFCN of the BTS.
6. All known nanoBTS units have a nominal transmit power of 23 dBm. If a `max_power_red` of 20 (dB) is configured, the resulting output power at the BTS Tx port is 23 - 20 = 3 dBm.

Note
The `nominal_power` setting does *not* influence the transmitted power to the BTS! It is a setting by which the system administrator tells the BSC about the nominal output power of the BTS. The BSC uses this as basis for calculations.

12.3 Example configuration for OsmoNITB with multi-TRX nanoBTS

Example 12.3 OsmoNITB configured for dual-TRX (stacked) nanoBTS

```
e1_input
 e1_line 0 driver ipa
network
 network country code 1
 mobile network code 1
 short name OpenBSC
 long name OpenBSC
 auth policy closed
 location updating reject cause 13
 encryption a5 0
 neci 1
 rrlp mode none
 mm info 0
 handover 0
 bts 0
  type nanobts
  band DCS1800
  cell_identity 0
  location_area_code 1
  training_sequence_code 7
  base_station_id_code 63
  ms max power 15
```

```
cell reselection hysteresis 4
rxlev access min 0
channel allocator ascending
rach tx integer 9
rach max transmission 7
ip.access unit_id 1800 0 ❶
oml ip.access stream_id 255 line 0
gprs mode none
trx 0
 rf_locked 0
 arfcn 871
 nominal power 23
 max_power_red 0
 rsl e1 tei 0
  timeslot 0
   phys_chan_config CCCH+SDCCH4
  timeslot 1
   phys_chan_config SDCCH8
  timeslot 2
   phys_chan_config TCH/F
  timeslot 3
   phys_chan_config TCH/F
  timeslot 4
   phys_chan_config TCH/F
  timeslot 5
   phys_chan_config TCH/F
  timeslot 6
   phys_chan_config TCH/F
  timeslot 7
   phys_chan_config TCH/F
trx 1
 rf_locked 0
 arfcn 873
 nominal power 23
 max_power_red 0
 rsl e1 tei 0
  timeslot 0
   phys_chan_config SDCCH8
  timeslot 1
   phys_chan_config TCH/F
  timeslot 2
   phys_chan_config TCH/F
  timeslot 3
   phys_chan_config TCH/F
  timeslot 4
   phys_chan_config TCH/F
  timeslot 5
   phys_chan_config TCH/F
  timeslot 6
   phys_chan_config TCH/F
  timeslot 7
   phys_chan_config TCH/F
```

❶ In this example, the IPA Unit ID is specified as `1800 0`. Thus, the first nanoBTS unit (`trx 0`) needs to be configured to 1800/0/0 and the second nanoBTS unit (`trx 1`) needs to be configured to 1800/0/1. You can configure the BTS unit IDs using the `ipaccess-config` utility included in OpenBSC.

Note
For building a multi-TRX setup, you also need to connect the TIB cables between the two nanoBTS units, as well as the coaxial/RF AUX cabling.

13 OsmoNITB HLR subsystem

As OsmoNITB is a fully autonomous system, it also includes a minimal/simplistic HLR and AUC. Compared to real GSM networks, it does not implement any of the external interfaces of a real HLR, such as the MAP/TCAP/SCCP protocol. It can only be used inside the OsmoNITB.

While functionally maintaining the subscriber database and authentication keys, it offers a much reduced feature set. For example, it is not possible to configure bearer service permission lists, or BAOC.

At this time, the only supported database back end for the OsmoNITB internal HLR/AUC is the file-based SQL database SQLite3.

13.1 Authorization Policy

Authorization determines how subscribers can access your network. This is unrelated to authentication, which verifies the authenticity of SIM cards that register with the network.

OsmoNITB supports three different authorization policies:

closed
: This mode requires subscribers to have a record with their IMSI in the HLR, and it requires that their status is set to `authorized 1`
This reflects the most typical operation of GSM networks, where subscribers have to obtain a SIM card issued by the operator. At the time the SIM gets issued, it is provisioned in the HLR to enable the subscriber to use the services of the network.

accept-all
: This policy accepts any and all subscribers that every try to register to the network. Non-existent subscribers are automatically and dynamically created in the HLR, and they immediately have full access to the network. Any IMSI can register, no matter what SIM card they are using in their phones.
This mode is mostly useful for lab testing or for demonstrating the lack of mutual authentication and the resulting security problems in the GSM system.

> **Note**
> As you do not know the Ki of dynamically created subscribers with SIM cards of unknown origin, you cannot use cryptographic authentication and/or encryption!

> **Caution**
> Never run a network in accept-all mode, unless you know exactly what you are doing. You are very likely causing service interruption to mobile phones in the coverage area of your BTSs, which is punishable under criminal law in most countries!

token
: This method was created for special-purpose configurations at certain events. It tries to combine the benefits of automatic enrollment with foreign IMSI while trying to prevent causing disruption to phones that register to the network by accident. This policy is currently not actively supported.

The currently active policy can be selected using the `auth policy (closed|accept-all|token)` at the `network` configuration node of the VTY.

13.2 Location Update Reject Cause

When a *Location Update Request* is to be rejected by the network (e.g. due to an unknown or unauthorized subscriber), the *Location Update Reject* message will contain a *Reject Cause*.

You can configure the numeric value of that cause by means of the `location updating reject cause <2-111>` command at the network node.

13.3 Querying information about a subscriber

Information about a specific subscriber can be obtained from the HLR by issuing `show subscriber` command.

For example, to display information about a subscriber with the IMSI 602022080345046, you can use the following command:

Displaying information about a subscriber

```
OpenBSC> show subscriber imsi 602022080345046
    ID: 1, Authorized: 1 ❶
    Name: 'Frank'
    Extension: 2342 ❷
    LAC: 1/0x1 ❸
    IMSI: 602022080345046
    TMSI: 4DB8B4D8
    Pending: 0
    Use count: 1
```

❶ Whether or not the subscriber is authorized for access

❷ OsmoNITB is often treated like a PBX, this is why phone numbers are called extensions

❸ The Location Area Code (LAC) indicates where in the network the subscriber has last performed a LOCATION UPDATE. Detached subscribers indicate a LAC of 0.

Subscribers don't have to be identified/referenced by their IMSI, but they can also be identified by their extension (phone number), their TMSI as well as their internal database ID. Example alternatives showing the same subscriber record are:

```
OpenBSC> show subscriber id 1
```

or

```
OpenBSC> show subscriber extension 2342
```

13.4 Enrolling a subscriber

A subscriber can be added to the network in different ways:

1. authorizing an auto-generated subscriber
2. manually creating a subscriber using VTY commands
3. manually creating subscriber by insert into SQL database by external program

13.4.1 Authorizing an auto-generated subscriber

If the `subscriber-create-on-demand` configuration option is set in the `nitb` VTY config node, then OsmoNITB will automatically create a subscriber record for every IMSI that ever tries to perform a LOCATION UPDATE with the network. However, those subscriber records are marked as "not authorized", i.e. they will not be able to use your network.

You can latter on *authorize* any such a subscriber using the `subscriber IMSI ...authorized 1` command at the VTY enable node.

Example: Authorizing an auto-generated subscriber

```
OpenBSC> enable
OpenBSC# configure terminal
OpenBSC(config)# nitb
OpenBSC(config-nitb)# subscriber-create-on-demand ❶
OpenBSC(config-nitb)# end
OpenBSC# ❷
OpenBSC# subscriber imsi 262420123456789 authorized 1 ❸
```

❶ We first ensure that `subscriber-create-on-demand` is active

❷ At this time we ensure that the MS with IMSI 262420123456789 performs a location update to our network, e.g. by powering up the associated phone followed by manual operator selection

❸ Here we authorize that ISMI

The above method implies that you know the IMSI stored on the SIM card of the subscriber that you want to to authorize. Unfortunately there is no easy/standard way to obtain the IMSI on most phones. If the phone has an AT-command interface, you may try `AT+CIMI`. You can also read the IMSI off the SIM using a PC-attached smart card reader.

Note
Contrary to classic GSM networks and for historic reasons, this behavior is the default behavior of OsmoNITB. For production networks with a closed subscriber base, it is strongly recommended to use the `no subscriber-create-on-demand` option at the `nitb` VTY config node.

13.4.2 Manually creating a subscriber from the VTY

You can manually add a subscriber to the HLR by VTY commands. To do so, yo will need to know at the minimum the IMSI of the subscriber.

Example: Create a new subscriber for IMSI 262429876543210

```
OpenBSC# subscriber create imsi 262429876543210
    ID: 3, Authorized: 0 ❶
    Extension: 22150 ❷
    LAC: 0/0x0 ❸
    IMSI: 262429876543210
    Expiration Time: Thu, 01 Jan 1970 01:00:00 +0100
    Paging: not paging Requests: 0
    Use count: 1
OpenBSC# subscriber imsi 262429876543210 authorized 1 ❹
OpenBSC# subscriber imsi 262429876543210 extension 23234242 ❺
OpenBSC# subscriber imsi 262429876543210 name Sub Scriber ❻
OpenBSC# show subscriber imsi 262429876543210 ❼
    ID: 3, Authorized: 1
    Name: 'Sub Scriber'
    Extension: 23234242
    LAC: 0/0x0
    IMSI: 262429876543210
```

```
    Expiration Time: Thu, 01 Jan 1970 01:00:00 +0100
    Paging: not paging Requests: 0
    Use count: 1
```

1. as you can see, a newly-created subscriber is not automatically authorized. We will change this in the next step.
2. the NITB has automatically allocated a random 5-digit extension (MSISDN)
3. Location Area Code 0 means that this subscriber is currently not registered on the network
4. Authorize the subscriber
5. Change the extension (MSISDN) to 23234242 (optional)
6. Give the subscriber a human-readable name (optional)
7. Review the content of your new subscriber record

Note
If you are running a network with A5 encryption enabled, you must also configure the secret key (Ki) of the SIM card in the HLR.

You can change further properties on your just-created subscriber as explained in Section 13.5.

13.4.3 Creating subscribers in the SQL database

In most applications, the network operator issues his own SIM cards, and the subscriber records corresponding to each SIM will be pre-provisioned by direct insertion into the SQL database. This is performed long before the SIM cards are issued towards the actual end-users.

This can be done by a custom program, the SQL schema is visible from the `.schema` command on the sqlite3 command-line program, and there are several scripts included in the OpenBSC source code, written in both Python as well as Perl language.

In case you are obtaining a starter kit with pre-provisioned SIM cards from sysmocom: They will ship with a HLR SQL database containing the subscriber records.

13.4.4 Provisioning SIM cards

In most applications, the operator obtains pre-provisioned SIM cards from a SIM card supplier.

If you prefer to provision the SIM cards yourself, you can use the pySim tool available from http://cgit.osmocom.org/cgit/pysim/. It has the ability to append the newly-provisioned SIM cards to an existing HLR database, please check its `--write-hlr` command line argument.

13.5 Changing subscriber properties

Once a subscriber exists in the HLR, his properties can be set interactively from the VTY. Modifying subscriber properties requires the VTY to be in the privileged (`enable`) mode.

All commands are single-line commands and always start with identifying the subscriber on which the operation shall be performed. Such identification can be performed by

- IMSI
- TMSI
- extension number
- ID (internal identifier)

13.5.1 Changing the subscriber phone number

You can set the phone number of the subscriber with IMSI 602022080345046 to 12345 by issuing the following VTY command from the enable node:

Changing the phone number of a subscriber

```
OpenBSC# subscriber imsi 602022080345046 extension 12345
```

13.5.2 Changing the subscriber name

The subscriber name is an internal property of OsmoNITB. The name will never be transmitted over the air interface or used by the GSM protocol. The sole purpose of the name is to make log output more intuitive, as human readers of log files tend to remember names easier than IMSIs or phone numbers.

In order to set the name of subscriber with extension number 12345 to "Frank", you can issue the following command on the VTY enable node: `subscriber extension 12345 name Frank`

The name may contain spaces and special characters. You can verify the modified subscriber record by issuing the `show subscriber extension 12345` command.

13.5.3 Changing the authorization status

As the HLR automatically adds records for all subscribers it sees, those that are actually permitted to use the network have to be authorized by setting the authorized property of the subscriber.

You can set the authorized property by issuing the following VTY command from the enable node:

Authorizing a subscriber

```
OpenBSC# subscriber extension 12345 authorized 1
```

Similarly, you can remove the authorized status from a subscriber by issuing the following command:

Un-authorizing a subscriber

```
OpenBSC# subscriber extension 12345 authorized 0
```

13.5.4 Changing the GSM authentication algorithm and Ki

In order to perform cryptographic authentication of the subscriber, his Ki needs to be known to the HLR/AUC. Furthermore, the authentication algorithm implemented on the SIM card (A3/A8) must match that of the algorithm configured in the HLR.

Currently, OsmoNITB supports the following authentication algorithms:

none
: No authentication is performed

xor
: Authentication is performed using the XOR algorithm (for test/debugging purpose)

comp128v1
: Authentication is performed according to the COMP128v1 algorithm

> **Warning**
> None of the supported authentication algorithms are cryptographically very strong. Development is proceeding to include support for stronger algorithms like GSM-MILENAGE. Please contact sysmocom if you require strong authentication support.

In order to configure a subscriber for COMP128v1 and to set his Ki, you can use the following VTY command from the enable node:

Configuring a subscriber for COMP128v1 and setting Ki

```
OpenBSC# subscriber extension 2342 a3a8 comp128v1 000102030405060708090a0b0c0d0e0f
```

14 Short Message Peer to Peer (SMPP)

The *Short Message Peer to Peer (SMPP) Protocol* [smpp-34] has been used for the communication with SMSCs. Osmocom implements version 3.4 of the protocol. Using this interface one can send MT-SMS to an attached subscriber or receive unrouted MO-SMS.

SMPP is served by the Osmocom MSC layer (both in the old OsmoNITB as well as the new OsmoMSC.

SMPP describes a situation where multiple ESMEs (External SMS Entities) interact with a SMSC (SMS Service Center) via the SMPP protocol. Each entity is identified by its System Id. The System ID is a character string which is configured by the system administrator.

OsmoNITB implements the SMSC side of SMPP and subsequently acts as a TCP server accepting incoming connections from ESME client programs.

Each ESME identifies itself to the SMSC with its system-id and an optional shared password.

14.1 Global SMPP configuration

There is a `smpp` vty node at the top level of the OsmoNITB configuration. Under this node, the global SMPP configuration is performed.

Use the `local-tcp-ip` command to define the TCP IP and port at which the OsmoNITB internal SMSC should listen for incoming SMPP connections. The default behaviour is to listen on all IPs (0.0.0.0), and the default port assigned to SMPP is 2775.

Use the `system-id` command to define the System ID of the SMSC.

Use the `policy` parameter to define whether only explicitly configured ESMEs are permitted to access the SMSC (`closed`), or whether any ESME should be accepted (`accept-all`).

Use the `smpp-first` command to define if SMPP routes have higher precedence than MSISDNs contained in the HLR (`smpp-first`), or if only MSISDNs found not in the HLR should be considered for routing to SMPP (`no smpp-first`).

14.2 ESME configuration

Under the `smpp` vty node, you can add any number of `esme` nodes, one for each ESME that you wish to configure.

Use the `esme NAME` command (where NAME corresponds to the system-id of the ESME to be configured) under the SMPP vty node to enter the configuration node for this given ESME.

Use the `password` command to specify the password (if any) for the ESME.

Use the `default-route` command to indicate that any MO-SMS without a more specific route should be routed to this ESME.

Use the `deliver-src-imsi` command to indicate that the SMPP DELIVER messages for MO SMS and the SMPP ALERT should state the IMSI (rather than the MSISDN) as source address.

Use the `osmocom-extensions` command to request that Osmocom specific extension TLVs shall be included in the SMPP PDUs. Those extensions include the ARFCN of the cell, the L1 transmit power of the MS, the timing advance, the uplink and dwnlink RxLev and RxQual, as well as the IMEI of the terminal at the time of generating the SMPP DELIVER PDU.

Use the `dcs-transparent` command to transparently pass the DCS value from the SMS Layer3 protocols to SMPP, instead of converting them to the SMPP-specific values.

Use the `route prefix` command to specify a route towards this ESME. Using routes, you specify which destination MSISDNs should be routed towards your ESME.

14.3 Example configuration snippet

The following example configuration snippet shows a single ESME *galactica* with a prefix-route of all national numbers stating with 2342:

```
smpp
 local-tcp-port 2775
 policy closed
 no smpp-first
 esme galactica
  password SoSayWeAll
  deliver-src-imsi
  osmocom-extensions
   route prefix national isdn 2342
```

14.4 Osmocom SMPP protocol extensions

Osmocom has implemented some extensions to the SMPP v3.4 protocol.

These extensions can be enabled using the `osmocom-extensions` VTY command at `esme` level.

The TLV definitions can be found in the `<osmocom/gsm/protocol/smpp34_osmocom.h>` header file provided by libosmocore.

14.4.1 RF channel measuremets

When the Osmocom SMPP extensions are enabled, we add the following TLVs to each SMPP DELIVER PDU:

TLV	IEI	Length (Octets)	Purpose
TLVID_osmo_arfcn	0x2300	2	GSM ARFCN of the radio interface
TLVID_osmo_ta	0x2301	1	Timing Advance on the radio interface
TLVID_osmo_ms_l1_txpwr	0x2307	1	Transmit Power of the MS in uplink direction
TLVID_osmo_rxlev_ul	0x2302	2	Uplink receive level as measured by BTS in dBm (int16_t)
TLVID_osmo_rxqual_ul	0x2303	1	Uplink RxQual value as measured by BTS
TLVID_osmo_rxlev_dl	0x2304	2	Downlink receive level as measured by MS in dBm (int16_t)
TLVID_osmo_rxqual_dl	0x2305	1	Downlink RxQual value as measured by MS

All of the above values reflect the **last measurement report** as recieved vi A-bis RSL from the BTS. It is thus a snapshot value (of the average within one 480ms SACCH period), and not an average over all the SACCH periods during which the channel was open or the SMS was received. Not all measurement reports contain all the values. So you might not get an TLVID_osmo_rxlev_dl IE, as that particular uplink frame might habe benn lost for the given snapshot we report.

14.4.2 Equipment IMEI

If we know the IMEI of the subscribers phone, we add the following TLV to each SMPP DELIVER PDU:

TLV	IEI	Length	Purpose
TLVID_osmo_imei	0x2306	variable	IMEI of the subscibers phone (ME)

15 MNCC for External Call Control

The 3GPP GSM specifications define an interface point (service access point) inside the MSC between the call-control part and the rest of the system. This service access point is called the MNCC-SAP. It is described in *3GPP TS 24.007* [3gpp-ts-24-007] Chapter 7.1.

However, like for all internal interfaces, 3GPP does not give any specific encoding for the primitives passed at this SAP.

The MNCC protocol has been created by the Osmocom community and allows to control the call handling and audio processing by an external application. The interface is currently exposed using Unix Domain Sockets. The protocol is defined in the `mncc.h` header file.

It is exposed by the Osmocom MSC layer (both in the old OsmoNITB as well as the new OsmoMSC.

OsmoNITB can run in two different modes:

1. with internal MNCC handler
2. with external MNCC handler

15.1 Internal MNCC handler

When the internal MNCC handler is enabled, OsmoNITB will switch voice calls between GSM subscribers internally and automatically based on the the subscribers *extension* number. No external software is required.

Note
Internal MNCC is the default behavior.

15.1.1 Internal MNCC Configuration

The internal MNCC handler offers some configuration parameters under the `mncc-int` VTY configuration node.

15.1.1.1 `default-codec tch-f (fr|efr|amr)`

Using this command, you can configure the default voice codec to be used by voice calls on TCH/F channels.

15.1.1.2 `default-codec tch-h (hr|amr)`

Using this command, you can configure the default voice codec to be used by voice calls on TCH/H channels.

15.2 External MNCC handler

When the external MNCC handler is enabled, OsmoNITB will not perform any internal call switching, but delegate all call-control handling towards the external MNCC program connected via the MNCC socket.

If you intend to operate OsmoNITB with external MNCC handler, you have to start it with the `-m` or `--mncc-sock` command line option.

At the time of this writing, the only external application implementing the MNCC interface compatible with the Osmocom MNCC socket is `lcr`, the Linux Call Router. More widespread integration of external call routing is available via the OsmoSIP-Connector.

15.3 MNCC protocol description

The protocol follows the primitives specified in 3GPP TS 04.07 Chapter 7.1. The encoding of the primitives is provided in the `mncc.h` header file in OsmoNITB's source tree, which uses some common definitions from `osmocom/gsm/mncc.h` (part of libosmocore.git).

However, Osmocom's MNCC specifies a number of additional primitives beyond those listed in the 3GPP specification.

The different calls in the network are distinguished by their callref (call reference), which is a unique unsigned 32bit integer.

15.3.1 MNCC_HOLD_IND

Direction: OsmoNITB → Handler

A *CC HOLD* message was received from the MS.

15.3.2 MNCC_HOLD_CNF

Direction: Handler → OsmoNITB

Acknowledge a previously-received *CC HOLD* message, causes the transmission of a *CC HOLD ACK* message to the MS.

15.3.3 MNCC_HOLD_REJ

Direction: Handler → OsmoNITB

Reject a previously-received *CC HOLD* message, causes the transmission of a *CC HOLD REJ* message to the MS.

15.3.4 MNCC_RETRIEVE_IND

Direction: OsmoNITB → Handler

A *CC RETRIEVE* message was received from the MS.

15.3.5 MNCC_RETRIEVE_CNF

Direction: Handler → OsmoNITB

Acknowledge a previously-received *CC RETRIEVE* message, causes the transmission of a *CC RETRIEVE ACK* message to the MS.

15.3.6 MNCC_RETRIEVE_REJ

Direction: Handler → OsmoNITB

Reject a previously-received *CC RETRIEVE* message, causes the transmission of a *CC RETRIEVE REJ* message to the MS.

15.3.7 MNCC_USERINFO_REQ

Direction: OsmoNITB → Handler

Causes a *CC USER INFO* message to be sent to the MS.

15.3.8 MNCC_USERINFO_IND

Direction: OsmoNITB → Handler

Indicates that a *CC USER-USER* message has been received from the MS.

15.3.9 MNCC_BRIDGE

Direction: Handler → OsmoNITB

Requests that the TCH (voice) channels of two calls shall be inter-connected. This is the old-fashioned way of using MNCC, historically required for circuit-switched BTSs whose TRAU frames are received via an E1 interface card, and works only when the TCH channel types match.

Note
Internal MNCC uses MNCC_BRIDGE to connect calls directly between connected BTSs or RNCs, in effect disallowing calls between mismatching TCH types and forcing all BTSs to be configured with exactly one TCH type and codec. This is a limitation that will probably remain for the old OsmoNITB. For the new OsmoMSC, the MNCC_BRIDGE command will instruct the separate OsmoMGW to bridge calls, which will be able to handle transcoding between different TCH as well as 3G (IuUP) payloads (but note: not yet implemented at the time of writing this). Hence an external MNCC may decide to bridge calls directly between BTSs or RNCs that both are internal to the OsmoMSC, for optimization reasons.

15.3.10 MNCC_FRAME_RECV

Direction: Handler → OsmoNITB

Enable the forwarding of TCH voice frames via the MNCC interface in OsmoNITB→Handler direction for the specified call.

15.3.11 MNCC_FRAME_DROP

Direction: Handler → OsmoNITB

Disable the forwarding of TCH voice frames via the MNCC interface in OsmoNITB→Handler direction for the specified call.

15.3.12 MNCC_LCHAN_MODIFY

Direction: Handler → OsmoNITB

Modify the current dedicated radio channel from signalling to voice, or if it is a signalling-only channel (SDCCH), assign a TCH to the MS.

15.3.13 MNCC_RTP_CREATE

Direction: Handler → OsmoNITB

Create a RTP socket for this call at the BTS/TRAU that serves this BTS.

15.3.14 MNCC_RTP_CONNECT

Direction: Handler → OsmoNITB

Connect the RTP socket of this call to the given remote IP address and port.

15.3.15 MNCC_RTP_FREE

Direction: Handler → OsmoNITB

Release a RTP connection for one given call.

15.3.16 GSM_TCHF_FRAME

Direction: both

Transfer the payload of a GSM Full-Rate (FR) voice frame between the OsmoNITB and an external MNCC handler.

15.3.17 GSM_TCHF_FRAME_EFR

Direction: both

Transfer the payload of a GSM Enhanced Full-Rate (EFR) voice frame between the OsmoNITB and an external MNCC handler.

15.3.18 GSM_TCHH_FRAME

Direction: both

Transfer the payload of a GSM Half-Rate (HR) voice frame between the OsmoNITB and an external MNCC handler.

15.3.19 GSM_TCH_FRAE_AMR

Direction: both

Transfer the payload of a GSM Adaptive-Multi-Rate (AMR) voice frame between the OsmoNITB and an external MNCC handler.

15.3.20 GSM_BAD_FRAME

Direction: OsmoNITB → Handler

Indicate that no valid voice frame, but a *bad frame* was received over the radio link from the MS.

16 Osmocom Control Interface

The VTY interface as described in Section 7 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.

16.1 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).

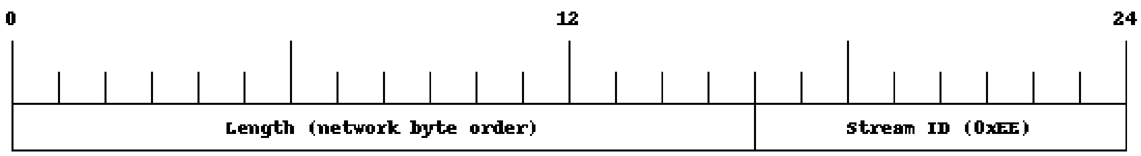

Figure 5: IPA header for control protocol

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

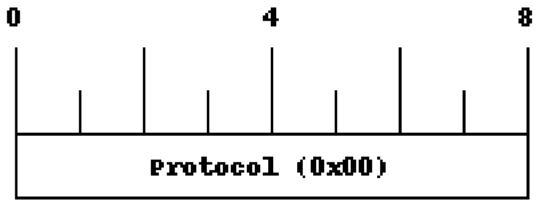

Figure 6: IPA extension header for control protocol

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.

16.1.1 GET operation

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

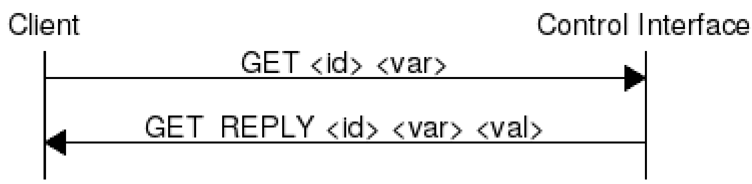

Figure 7: Control Interface GET operation (successful outcome)

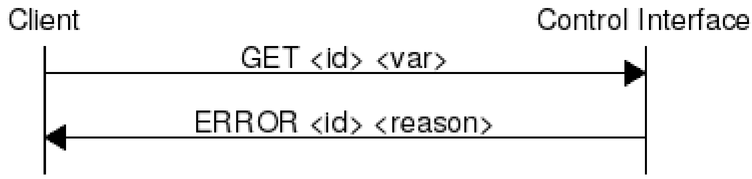

Figure 8: Control Interface GET operation (unsuccessful outcome)

16.1.2 SET operation

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

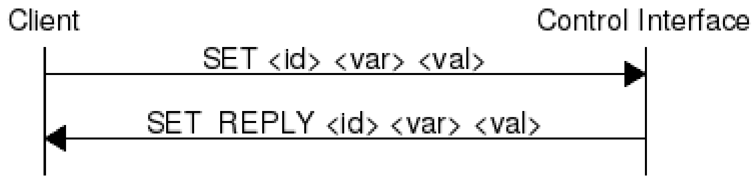

Figure 9: Control Interface SET operation (successful outcome)

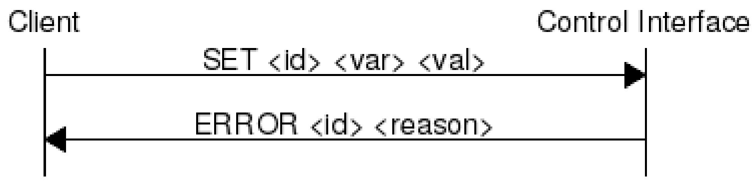

Figure 10: Control Interface SET operation (unsuccessful outcome)

16.1.3 TRAP operation

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

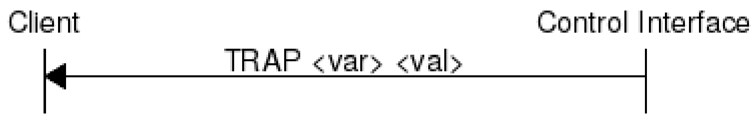

Figure 11: Control Interface TRAP operation

16.2 Common variables

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

Table 6: Variables available over control interface

Name	Access	Value	Comment
counter.*	RO		Get counter value.
rate_ctr.*	RO		Get rate counter value.

Those read-only variables allow to get value of arbitrary counter or rate counter using its name e. g. "counter.net.sms.submitted" or "rate_ctr.per_hour.nat.bsc.sccp.conn". Of course for that to work the program in question have to register corresponding counter names using libosmocore functions. Note the difference between counter and rate_ctr access format: in case of rate_ctr the counter 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.

16.3 Control Interface python example: `bsc_control.py`

In the `openbsc.git` repository, there is an example python script called `openbsc/contrib/bsc_control.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.

16.3.1 Setting a value

Example: Use bsc_control.py to set the short network name of OsmoNITB

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

16.3.2 Getting a value

Example: Use bsc_control.py to get the mnc of OsmoNITB

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

16.3.3 Listening for traps

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

Example: Using bsc_control.py to listen for traps:

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

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

17 Cell Broadcast

Normally, all user plane data in GSM/GPRS networks are sent in point-to-point channels from the network to the user. Those are called "dedicated" radio channels which exist between the network and one given phone/subscriber at a time.

Cell Broadcast is an exception to that rule. It permits user data (so-called SMS-CB data) to be broadcast by the network in a way that can be received by all phones in the coverage area of the given BTS simultaneously.

More high-level information can be found at https://en.wikipedia.org/wiki/Cell_Broadcast and the related specification is [?].

17.1 Use Cases

Cell Broadcast was used for various different use cases primarily in the 1990ies and early 2000s, including

- advertisement of the GPS position of the cell tower you're currently camping on
- advertisement of the calling codes of your current "home zone", i.e. a "lower cost short distance" call zone travelling with you as you roam around.

More recently, SMS-CB is seeing some uptake by various disaster warning systems, such as

- CMAS (Commercial Mobile Alert System), later renamed to WEA (Wireless Emergency Alerts) in the US.
- EU-Alert in the European union
- Messer Ishi (Rocket Alert) in Israel
- ETWS (Earthquake and Tsunami Warning System) in Japan
- KPAS (Korean Public Alert System)

17.2 Osmocom Cell Broadcast support

- OsmoBTS implements the "SMS BROADCAST COMMAND" Message in RSL according to Section 8.5.8 of 3GPP TS 08.58
- OsmoNITB and OsmoBSC implement a VTY command `bts <0-255> smscb-command <1-4> HEXSTRING` to send a given hex-formatted cell broadcast message to a specified BTS

17.2.1 What's missing

What's missing (for production operation in larger networks)

- mechanism to broadcast one (set of) cell broadcast messages from the BSC to multiple/all BTSs, rather than one BTS individually
- OsmoBTS reporting of current CBCH load
- BSC scheduler scheduling multiple alternating sets of CBCH messages based on the current CBCH load reported by BTS
- external interface from BSC to a Cell Broadcast Center (CBC), e.g. according to 3GPP TS 48.049
- an Osmocom implementation of the Cell Broadcast Center (OsmoCBC) which can manage and distribute messages to multiple BSCs and which has an external interface by which cell-broadcast can be entered into the network

If you would like to contribute in any of those areas (by means of code or funding), please reach out to us any time.

17.3 Message Structure

- Each message has a meximum of 15 pages
- Each page is 82 bytes of data, resulting in 93 characters in GSM 7-bit default alphabet
- Messages are broadcast on logical channels (more like an address)
- Subscribers can activate/deactivate selective addresses

18 Abis/IP Interface

18.1 A-bis Operation & Maintenance Link

The GSM Operation & Maintenance Link (OML) is specified in 3GPP TS 12.21 and is used between a GSM Base-Transceiver-Station (BTS) and a GSM Base-Station-Controller (BSC). The default TCP port for OML is *3002*. The connection will be opened from the BTS to the BSC.

Abis OML is only specified over E1 interfaces. The Abis/IP implementation of OsmoBTS and OsmoBSC extend and/or deviate from the TS 12.21 specification in several ways. Please see the *OsmoBTS Abis Protocol Specification* [osmobts-abis-spec] for more information.

18.2 A-bis Radio Signalling Link

The GSM Radio Signalling Link (RSL) is specified in 3GPP TS 08.58 and is used between a GSM Base-Transceiver-Station and a GSM Base-Station-Controller (BSC). The default TCP port for RSL is *3003*. The connection will be opened from the BTS to BSC after it has been instructed by the BSC.

Abis RSL is only specified over E1 interfaces. The Abis/IP implementation of OsmoBTS and OsmoBSC extend and/or deviate from the TS 08.58 specification in several ways. Please see the *OsmoBTS Abis Protocol Specification* [osmobts-abis-spec] for more information.

18.3 Locate Abis/IP based BTS

We can use a tool called abisip-find to be able to find BTS which is connected in the network. This tool is located under: *./openbsc/openbsc/src/ipaccess*

18.3.1 abisip-find

abisip-find is a small command line tool which is used to search and find BTS devices in your network (e.g. sysmoBTS, nanoBTS).

It uses broadcast packets of the UDP variant of the Abis-IP protocol on port 3006, and thus will find any BTS that can be reached by the all-network broadcast address 255.255.255.255

When program is started it will print one line for each BTS it can find.

Example: using abisip-find to find BTS in your network

```
$ ./abisip-find
abisip-find (C) 2009 by Harald Welte
This is FREE SOFTWARE with ABSOLUTELY NO WARRANTY

you might need to specify the outgoing
network interface, e.g. ``abisip-find eth0``
Trying to find ip.access BTS by broadcast UDP...
```

```
MAC_Address='24:62:78:01:02:03' IP_Address='192.168.0.171' Serial_Number='123'
Unit_ID='sysmoBTS 1002'

MAC_Address='24:62:78:04:05:06' IP_Address='192.168.0.182' Serial_Number='456'
Unit_ID='sysmoBTS 1002'

MAC Address='00:01:02:03:04:05' IP Address='192.168.100.123' Unit ID='65535/0/0'
Location_1='' Location 2='BTS_NBT131G'  Equipment Version='165a029_55'
Software Version='168a302_v142b13d0' Unit Name='nbts-00-02-95-00-4E-B3'
Serial Number='00123456'

^C
```

You may have to start the program as a root:

```
$ sudo ./abisip-find eth0
```

18.4 Deploying a new nanoBTS

A tool called ipaccess-config can be used to configure a new ip.access nanoBTS.

18.4.1 ipaccess-config

This program is very helpful tool which is used to configure Unit ID and Primarly OML IP. You can find this tool under: *./openbsc/openbsc/src/ipaccess*

Example: using ipaccess-config to configure Unit ID and Primarly OML IP of nanoBTS

```
$ ./ipaccess-config -u 1801/0/0❶  10.9.1.195❷ -o 10.9.1.154❸

ipaccess-config (C) 2009-2010 by Harald Welte and others
This is FREE SOFTWARE with ABSOLUTELY NO WARRANTY

Trying to connect to ip.access BTS ...
abis_nm.c:316 OC=SITE-MANAGER(00) INST=(ff,ff,ff) STATE CHG:
OP_STATE=Disabled AVAIL=Not installed(07)
abis_nm.c:316 OC=BTS(01) INST=(00,ff,ff) STATE CHG:
OP_STATE=Disabled AVAIL=Not installed(07) ADM=Locked
abis_nm.c:316 OC=BASEBAND-TRANSCEIVER(04) INST=(00,00,ff) STATE CHG:
OP_STATE=Disabled AVAIL=Not installed(07) ADM=Locked
OML link established using TRX 0
setting Unit ID to '1801/0/0'
setting primary OML link IP to '10.9.1.154'
abis_nm.c:316 OC=CHANNEL(03) INST=(00,00,00) STATE CHG:
OP_STATE=Disabled AVAIL=Not installed(07) ADM=Locked
...
abis_nm.c:2433 OC=BASEBAND-TRANSCEIVER(04) INST=(00,00,ff) IPACCESS(0xf0):
SET NVATTR ACK
Set the NV Attributes.
```

❶ Unit ID

❷ IP address of the NITB

❸ IP address of the nanoBTS

19 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

GFDL
GNU Free Documentation License; a copyleft-style Documentation License

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 7: TCP/UDP port numbers

L4 Protocol	Port Number	Purpose	Software
UDP	2427	MGCP GW	osmo-bsc_mgcp
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
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	ggsn (OpenGGSN)
TCP	4258	telnet (VTY)	osmo-hlr
TCP	4259	Control Interface	osmo-hlr
TCP	4260	telnet (VTY)	ggsn (OpenGGSN)
TCP	4261	telnet (VTY)	osmo-mgw
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.0.1.0.1 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.