Amarisoft Off The Shelf service

Version: 2025-05-21

Table of Contents

1 Introduction 1

1.1 General description 1
1.2 Installation 1

2 Using service 2

2.1 Service management 2
2.1.1 Status 2
2.1.2 Stop 2
2.1.3 Start 2
2.1.4 Disable 2
2.1.5 Enable 2
2.2 Access to software monitors 2
2.2.1 Select component 3
2.2.2 Exit screen 4

3 Configuration 5

3.1 File 5

4 Parameters 6

5 Miscellaneous 8

5.1 Logs 8
5.2 Contact 8

6 Additional Information 9

1 Introduction

1.1 General description.

This document describes how to configure and use Amarisoft OTS service.
OTS service is a systemd or init.d Linux service allowing to start/stop and manage automatically Amarisoft software components.

1.2 Installation

You need to use the `install.sh` provided in your Amarisoft software release tarball and answer yes to its installation during the process:

```
1) Configuration
  You can exit install script during this step, nothing will be changed
  until next step
  * Do you want to install LTE automatic service ?
    Package screen may be installed. [Yn] y
```

Enabling it means enabling Linux service by default:

```
    - Do you want to enable LTE automatic service ? [yN] n
```

2 Using service

2.1 Service management

2.1.1 Status

You can check the LTE service status this way:

```
service lte status
```

The command will return "`active (running)`" status if service is running

2.1.2 Stop

You can stop all LTE components this way:

```
service lte stop
```

2.1.3 Start

You can start them again this way:

```
service lte start
```

2.1.4 Disable

You may also prevent them to start at boot time:

```
systemctl disable lte
```

NB: lte service remains enable until next reboot
NB2: this command is not available on Ubuntu version <= 14

2.1.5 Enable

You may enable service at boot time this way:

```
systemctl enable lte
```

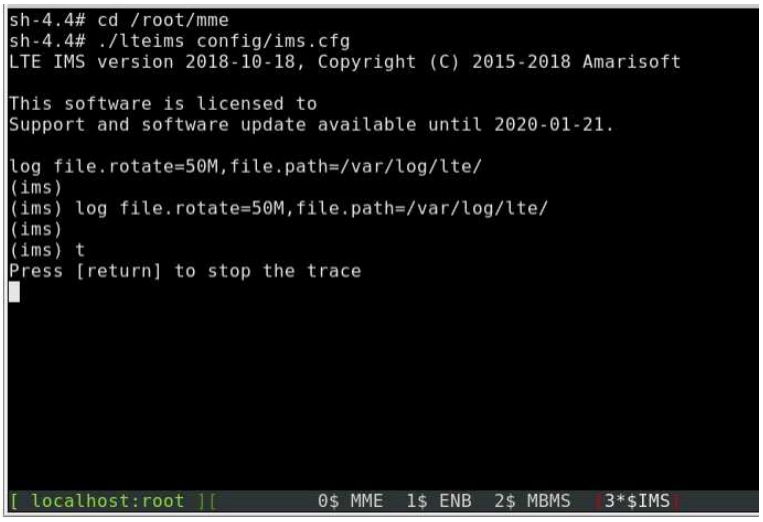
NB: lte service remains disable until next reboot
NB2: this command is not available on Ubuntu version <= 14

2.2 Access to software monitors

Once you are logged on your callbox, you can access software components (eNB, MME, IMS or MBMSGW) using screen command:

```
screen -x lte
```

This will connect you to different component monitor.



Next sections show you basic methods. For more information please refer to `screen` documentation (`https://www.gnu.org/software/screen/manual/screen.html`).

2.2.1 Select component

Each component monitor is inside a window. You can switch from a window to another with the command:

```
ctrl+a <window index>
```

Where window index is:

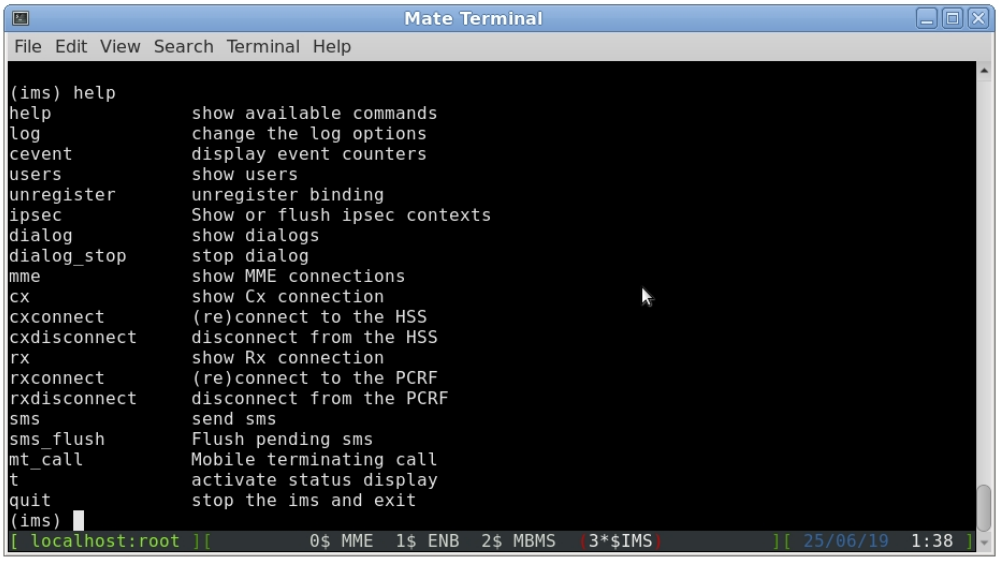
- 0 MME
- 1 eNB
- 2 MBMSGW
- 3 IMS

> Note: press simultaneously `CTRL` key and `a` key, release them, then press number key.

You can also switch to next window:

```
ctrl+a <space>
```

Each component screen offers a list of commands that can be used either to get status or trigger action. Each of them are documented in the component documentations (example lteenb.pdf) or inline with the "help" command



2.2.2 Exit screen

`ctrl+a d`

3 Configuration

3.1 File

The LTE automatic service uses `/root/ots/config/ots.cfg` for its configuration.
The format is shell.

The default configuration file is generated during installation.
To use your own configuration we recommend to create a new file and change `/root/ots/config/ots.cfg` symbolic link to point to your file.
Else, your changes will be overriden at next software install/upgrade.

Example:
Create a my-ots.cfg file and put the following inside:

```
# Include default configuration
source ots.default.cfg

# Add your custom config
MME_CONFIG_FILE=/root/mme/config/my-mme.cfg
```

Then:

```
cd /root/ots/config
rm -f ots.cfg
ln -s my-ots.cfg ots.cfg
```

4 Parameters

The configuration variables are:

`ERROR_DELAY`
: Time in seconds before restarting any software component badly stopped.

`LOG_FILE` Log file of the lte service.

`LOG_FILE_SIZE`
: Automatic service log file size for rotation (K, M, G units accepted).

`LOG_PATH` Path where to store rotated log.

`LOG_SIZE` Components log file size threshold for rotation (K, M, G units accepted). If log file size exceed this value, file will be rotated to `LOG_PATH` if set or to same directory.

`LOG_COUNT`
: Components log file count threshold for rotation (K, M, G units accepted). If number of dumped logs in file size exceed this value, file will be rotated to `LOG_PATH` if set or to same directory.

`LOG_PERSISTENT_SIZE`
: Maximum amount of logs stored in `LOG_PATH`. When directory reaches this threshold, oldest log file will be deleted.

`LOG_PERSISTENT_COUNT`
: Maximum number of log files stored in `LOG_PATH`. When directory reaches this threshold, oldest log file will be deleted.

`LOG_POLL_DELAY`
: Time in seconds between two checks for cleaning `LOG_PATH`.

`LOG_GZIP` Set it to 1 to compress log file when rotated.

`LOG_BACKUP_ON_STOP`
: Set it to n to avoid log backup (to LOG_PATH) on component stop.

`COMPONENTS`
: List of component IDs to start.

Each component can be configured with the following variables where `<COMP>` is the `ID` of the component and must be unique.

`<COMP>_PATH`
: Path of the component binary. If not found or not set, component won't be started.

`<COMP>_CONFIG_FILE`
: Config file to use for this component. Relative to `XXX_PATH`.

`<COMP>_TITLE`
: Title to display in `screen` component. `ID` by default.

`<COMP>_INIT`
: `lte_init.sh` command line parameter for component.

`<COMP>_AUTOSTART`
: If set to 1 or not defined, component will be started automatically by service, else, you need to start it manually.

`<COMP>_TYPE`
: Component flavour. Can be `MME`, `ENB`, `IMS`, `MBMSGW`, `LICENSE`...

`<COMP>_WIN`
: Component window index inside screen session.

`<COMP>_CONFIG_FILE`
: Component configuration file

`<COMP>_OUTPUT_FILE`
: If set, defineds a filename where to dump both `stdout` and `stderr` of the component process.
If not set, only `stderr` will be dumped to `/tmp/.stderr.<COMP>`

`<COMP>_SCRIPT`
: If set, defines a shell script that will be executed on each component state change. Arguments are <COMP> <STATE> [<ERROR>] where state can be:
  - starting: before executing component binary
  - started: after component binary has started
  - error: when any error occurred (In this case the third argument will be filed with error message)
  - stop: after successful component stop

  This can be used to perform any action before component start by using the state starting.
  Note that the script is executed synchronously and may delay or block component startup.

IDs generated by installation process are the same as type.

Those variables are used by `ltestart.sh` script.

For instance, if you want to setup two `MME` instance, use the following:

```
MME_PATH=/root/mme
MME_TYPE=MME
...

MME2_PATH=/root/mme
MME2_TYPE=MME
...

COMPONENTS+=" MME MME1"
```

5 Miscellaneous

5.1 Logs

Logs will be backed up in `/var/log/lte` directory.

The `/tmp/lte.log` file is log for service.

5.2 Contact

- Our Extranet site is located at `extranet.amarisoft.com`. This site would give you access to our documentation and new releases.
- For all technical issues, you can create a ticket describing your problem on our support site at `support.amarisoft.com`. Please note that you need to have an active account in our Extranet in order to be able to login to our support site. Your credentials are the same as the ones you use to access Extranet.
- our FTP server is located at `ftp.amarisoft.com`. This server should be used to exchange big logs. Your credentials are the same as the ones you use to access Extranet. Please note that you need to use an FTP client supporting SSL such as Filezilla.
- For any request concerning license delivery or addition of new accounts, please send an email to `delivery@amarisoft.com`.

6 Additional Information

For any technical issue, please raise a ticket from our support site at `https://support.amarisoft.com/`.
To learn more about our technology and solutions, e-mail us at `customer@amarisoft.com` or visit `https://www.amarisoft.com`.