The trx-control User Guide

Marc Balmer HB9SSB <info@hb9ssb.ch>

Table of Contents

Installing trx-control

Using Prebuilt Binary Packages

The easiest and fastest way to install trx-control is by installing a prebuilt binary package for your Linux distribution.

Debian Based Systems

First install the GPG public key that is used to sign the repository:

```
# curl -o /etc/apt/trusted.gpg.d/trx-control.asc \
    https://trx-control.msys.ch/pub/repos/trx-control.asc
```

Then add a repository file, substitute <codename> by the codename of your Linux distribution (e.g. jammy, bookworm etc.):

```
# echo deb https://trx-control.msys.ch/pub/repos/apt/ <codename> stable > \
    /etc/apt/sources.list.d/trx-control.list
```

Distribution	Repository name	Available architectures
Debian 13 (Sid, Unstable)	trixie	x86_64, aarch64
Debian 12	bookworm	x86_64, aarch64
Debian 11	bullseye	x86_64, aarch64
Debian 10	buster	x86_64, aarch64
Ubuntu 23.10	mantic	x86_64
Ubuntu 23.04	lunar	x86_64
Ubuntu 22.04	jammy	x86_64, aarch64
Ubuntu 20.04	focal	x86_64, aarch64

Your are now all set, update the package cache and install trx-control:

```
# apt update
# apt install trx-control
```

Red Hat Based Systems

Install the trx-control repository package using dnf for your system from https://trx-control.msys.ch/pub/repos/yum/<distribution>/noarch/trx-control-repo-latest.noarch.rpm, e.g. for

AlmaLinux 9 use:

```
# dnf install https://trx-control.msys.ch/pub/repos/yum/alma-9/noarch/trx-control-repo-latest.noarch.rpm
```

Distribution	Repository name	Available architectures
AlmaLinux 9	alma-9	x86_64, aarch64
AlmaLinux 8	alma-8	x86_64, aarch64
Rocky Linux 9	rocky-9	x86_64, aarch64
Rocky Linux 8	rocky-8	x86_64, aarch64
CentOS 7	centos-7	x86_64, aarch64

Then install trx-control:

```
# dnf install trx-control --refresh
```

SUSE Based Systems

Install the trx-control repository package using zypper for your system from https://trx-control.msys.ch/pub/repos/yum/<distribution>/noarch/trx-control-repo-latest.noarch.rpm

Distribution	Repository name	Available architectures
OpenSUSE Tumbleweed	opensuse-tumbleweed	x86_64, aarch64
OpenSUSE Leap 15.5	opensuse-leap-15.5	x86_64, aarch64

Then install trx-control:

```
# zypper install trx-control
```

Manual Package Download

The packages can be downloaded from https://trx-control.msys.ch/pub/repos

There are three repositories:

apt/	Debian based systems
yum/	Red Hat based systems
zypp/	SUSE based systems

Build From Source

To build trx-control from source code, you need to have gcc and make installed on your system as well some development packages. In general, any linux system set up for development purposes should do it.

Build Dependencies

The following build dependencies must be installed before trx-control can be compiled:

Red Hat Based Systems	Debian Based Systems	Suse Based Systems	Purpose / Comments
libbsd-devel	libbsd-dev	libbsd-devel	general use
motif-devel	libmotif-dev	motif-devel	xqrg
openssl-devel	libssl-dev	libopenssl-devel	trxd
readline-devel	libreadline-dev	readline-devel	trxctl
libyaml-devel	libyaml-dev	libyaml-devel	YAML Lua module
libcurl-devel	libcurl4-openssl-dev	libcurl-devel	cURL Lua module
expat-devel	libexpat1-dev	libexpat-devel	Expat Lua module
postgresql16-devel	libpq-dev postgresql-server-dev-16	postgresql16-devel	PostgreSQL Lua module
sqlite-devel	libsqlite3-dev	sqlite3-devel	SQLite Lua module

After installing the build dependencies, clone the source code using git:

```
$ git clone https://github.com/hb9ssb/trx-control.git
```

Then build trx-control using make:

```
$ cd trx-control
$ make
$ sudo make install
```

This will build all of trx-control, install the binaries and support files as well as the manual pages trxctl(1), xqrg(1), trxd(8), and, trx-control(7).

The trxd(8) Configuration file

trxd(8) reads its configuration from the file /etc/trxd.yaml by default. A different path can be specified with the -c option on the command line. The configuration file must be in YAML-format (see https://yaml.org for details).

A sample configuration file can be found in /usr/share/trxd/trxd.yaml.

In the configuration file you can set global defaults like the address and port where trxd(8) should listen for incoming client connections. Transceivers are listed in the configuration by choosing an arbitrary name and specifying the correct driver. One transceiver can be designated as the default transceiver.

This configuration file will be processed using the yaml Lua module, so the local tags mentioned in https://lua.msys.ch/lua-module-reference.html#_a_note_on_yaml_tags can be used.

Example configuration

```
# Example trxd(8) config file.  Note this in YAML-format.

# Run in the background
no-daemon: false

# Listen on localhost:14285 for incoming connections
bind-address: localhost
listen-port: 14285

# Listen on port 14290 for WebSocket connections
websocket:
  bind-address: localhost
  listen-port: 14290

  # If a certificate path is defined, wss is used instead of ws
  # certificate: server.pem
  handshake: trx-control

# Log incoming connection using syslog
log-connections: true

# trxd shall run as trxd:trxd
user: trxd
group: trxd

# Store the PID of the running trxd process in trxd.pid
pid-file: trxd.pid

# The list of our transceivers
trx:
  ft-897:
```

```
    device: /dev/ttyUSB2
    speed: 38400
    driver: yaesu-ft-897

    # This is the default transceiver in case the client does not explicitely
    # select a transceiver by name.
    default: true

  dummy:
    device: /dev/null
    driver: dummy-trx

  ft-710:
    device: /dev/ttyUSB0
    speed: 38400
    driver: yaesu-ft-710

extensions:
  ping:
    script: ping

 keepalive:
    script: keepalive
    callable: false
      configuration:
        timeout: 300

  logbook:
    script: logbook
      configuration:
        connStr: dbname=logbook
        datestyle: German

  dxcluster:
    script: dxcluster
      configuration:
        host: wr3d.dxcluster.net
        port: 7300
        callsign: MYCALLSIGN

  sotacluster:
    script: dxcluster
      configuration:
        host: cluster.sota.co.uk
        port: 7300
        callsign: MYCALLSIGN
        source: sotacluster

  qrz:
    script: qrz
      configuration:
```

```
        username: MYCALLSIGN
        passwort: sicrit

  tasmota:
    script: tasmota
      configuration:
        address: 192.168.4.1
```