micro systems
trx-control

The trx-control User Guide

Marc Balmer HB9SSB <info@hb9ssb.ch>

Table of Contents

Installing trx-control

Prerequisites

Whether you install trx-control from prebuilt binary packages or build it from source codes, you must first install the PostgreSQL repository as PostgreSQL is used for all database uses (e.g. the logbook or memory system).

For Debian based systems, please install the PostgreSQL repository by following the instructions found on https://apt.postgresql.org.

For Red Hat based systems, please install the PostgreSQL repository by following the instructions found on https://yum.postgresql.org.

Install 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 24.04	noble	x86_64, aarch64
Ubuntu 23.10	mantic	x86_64
Ubuntu 23.04	lunar	x86_64

Distribution	Repository name	Available architectures
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
Fedora 40	fedora-40	x86_64, aarch64
Fedora 39	fedora-39	x86_64, aarch64
Fedora 38	fedora-38	x86_64, aarch64
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.

> Install the PostgreSQL repository first as some PostgreSQL related packages are needed to build trx-control. For Debian based systems, see https://apt.postgresql.org, for Red Hat based systems, see https://yum.postgresql.org for instructions.

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
avahi-devel	libavahi-client-dev	libavahi-devel	Avahi (mDNS) client
bluez-libs-devel	libbluetooth-dev	bluez-devel	Bluetooth communication
libbsd-devel	libbsd-dev	libbsd-devel	general use
motif-devel	libmotif-dev	motif-devel	xqrg
openssl-devel	libssl-dev	libopenssl-devel	trxd

Red Hat Based Systems	Debian Based Systems	Suse Based Systems	Purpose / Comments
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.

Connecting To Transceivers

Transceivers can be connected either over a serial line or via Bluetooth. Whether a serial line or Bluetooth is used makes a difference in device naming:

Serial line device names are an absolute path to a device, e.g. `/dev/ttyUSB0`, and they have the optional `speed` setting to indicate the bit rate.

Bluetooth device names, however, are six pairs of hex digits separated by a colon, e.g. `01:23:45:67:89:AB`, and they have the optional `channel` setting to indicate the RFCOMM channel to be used.

Example configuration

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

# Run in the background
no-daemon: false

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

# Listen on all interfaces for WebSocket connections
websocket:
  bind-address: 0.0.0.0
  listen-port: 14290
  path: trx-control

  # If a certificate path is defined, wss is used instead of ws
  # certificate: server.pem
```

```yaml
  # If you don't want to announce the trx-control service using Avahi (mDNS),
  # set announce to false.  The default is to announce the service as
  # _trx-control._tcp:
  announce: true

# 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

# Decode NMEA sentences from a GPS/Glonass/Baidu etc. receiver
nmea:
  device: /dev/ic-705-nmea
  speed: 9600

# 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

  # An ICOM IC-705 connected over Bluetooth serial (must be paired first)
  ic-705:
    device: 01:23:45:67:89:AB
    channel: 3
    driver: icom-ic-705
    # Optionally set the controller and transceiver address.
    configuration:
      controllerAddress: 0xe0
      transceiverAddress: 0xa4

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

extensions:
```

```yaml
  ping:
    script: ping

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

  memory:
    script: memory
    configuration:
      connStr: dbname=trx-contol
      datestyle: German

  logbook:
    script: logbook
    configuration:
      connStr: dbname=trx-contol
      datestyle: German

  cloudlog:
    script: cloudlog
    configuration:
      url: https://cloudlog.xyz.com/index.php/api
      apiKey: xxxxxxxxxxxx

  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

  hamqth:
    script: hamqth
    configuration:
      username: MYCALLSIGN
      password: sicrit

  qrz:
    script: qrz
    configuration:
```

```
      username: MYCALLSIGN
      password: sicrit

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