Guide to Retroshare Tor or I2P routing Update

Some quick notes to supplement the existing Help Guides for setting up Retroshare Regular and Hidden Nodes routed via Tor and/or I2P.

Snapshot in this update
1) Point out the torrc exit node command is not needed for Regular and Hidden Node Retroshare Tor routing with torrc examples.
2) Steps, guides for setting up multiple Tor binary folders each using a different Socks Port for concurrent Tor applications to enable different specific listening ports, Tor proxys.
3) Guide for setting up a I2P 4/4a/5 client tunnel and need to turn off unused I2P default tunnels.

Retroshare Regular nodes using the optional Tor proxy client does not need nor use any Tor Exit Nodes. Unless you are using the same Tor binary for applications which require a Exit Node to clearnet such as with a FeedReader RSS server or website, do not include Exit Node commands in your Tor torrc file.

Retroshare Regular Node using the optional Tor Proxy client Working Torrc Example (adjust the file paths for your needs, system)

```
# This file was generated by Tor; if you edit it, comments will not be preserved
# The old torrc file was renamed to torrc.orig.1 or similar, and Tor will ignore it

DataDirectory /usr/local/etc/tor
GeoIPFile /usr/local/etc/tor/geoip
GeoIPv6File /usr/local/etc/tor/geoip6
SocksPort 9050
```

Retroshare Hidden nodes routed solely to and inside the Tor Network should not have any Exit Node commands in your Tor torrc file.

Working example of a Retroshare Hidden node operating as a Tor Hidden Service torrc file

```
# This file was generated by Tor; if you edit it, comments will not be preserved
# The old torrc file was renamed to torrc.orig.1 or similar, and Tor will ignore it

DataDirectory /usr/local/etc/tor
GeoIPFile /usr/local/etc/tor/geoip
GeoIPv6File /usr/local/etc/tor/geoip6
HiddenServiceDir /home/name/hideserv
HiddenServicePort 11040 127.0.0.1:13080
SocksPort 9050
```

Running Multiple concurrent applications routing each through the tor network as a Tor proxy client or as a Tor hidden service.

Tor purists run separate tor binary routers for each application that would be concurrently running. This helps provide greater stability and connectivity with the tor router provided you use a different SocksPort for tor with each separate tor folder. If you are running concurrent Tor routed applications all listening to the same SocksPort and expecting them to correctly operate, you are doing yourself a disservice. You have everything to gain by running separate tor binary’s with a different Tor SocksPort for each application you would concurrently run (at the same time).

To bootup tor as it is exampled and explained below, you must always use a custom command telling tor where to explicitly find the tor binary torrc command file. You bootup tor in these additional folders using the tor -f path/torrc command.

The Tor Browser Bundle available for all platforms which Retroshare supports (Windows,Linux,Mac) provides a quick and easy method to obtain the files you need to copy into a custom tor user folder that is set to another SockPort. The Tor Browser Bundle uses a client listening SockPort of 9100 which doesn't interfer with the standard tor file system installed tor binary running on SockPort 9050. The Tor Network uses Port 9000 and some systems use a Sock Control Port 9051 so we'll create additional tor binary folders setup for a SockPort between 9052-9099.

I prefer copying the needed files from The Tor Browser Bundle because I generally run my system file area installed tor binary with a Tor Hidden Service and the files are automatically adjusted for the internal Tor Network and caching for the Tor Hidden Service. The Tor Browser Bundle files do not have those which could cause problems creating new network circuits outside the Tor Network and you do not have to deal with the file permissions issues inherent with the tor binary installed files.

Example for setting up a user tor binary folder setup for a SockPort 9052

Create a new user folder for your additional tor binary and support files.

```
mkdir tor-9052
copy the tor binary from the Tor Browser Bundle
cd /home/name/Downloads/tor-browser_en-US/Browser/TorBrowser/Tor
tor
cp tor /home/name/tor-9052
```

```
copy the tor support files from the Tor Browser Bundle
cd /home/name/Downloads/tor-browser_en-US/Browser/TorBrowser/Data/Tor
copy all files including geoip,geoip6,torrc to your /tor-9052 folder
```

With a text editor, edit and save your torrc file to reflect your new paths to DataDirectory, GeoIPFile, GeoIPv6File and add SockPort 9052

```
DataDirectory /home/name/tor-9052
GeoIPFile /home/name/tor-9052/geoip
GeoIPv6File /home/name/tor-9052/geoip6
#ExitNodes {us} <-- Uncomment only if you need a exit node operating
#StrictNodes 1 <-- Uncomment only if you need a exit node operating
SocksPort 9052
```

To bootup your new user tor router with its custom SocksPort 9052
open a terminal, change to your new tor-9052 folder

```
cd ~/tor-9052
Always run this tor router using this example
$ tor -f /home/name/tor-9052/torrc
```

Expected successful bootup of your new tor SocksPort 9052 should be similar to the following example

```
May 31 19:20:23.240 [notice] Tor v0.2.7.6 running on Linux with Libevent 2.0.16-
stable, OpenSSL 1.0.1 and Zlib 1.2.3.4.
May 31 19:20:23.241 [notice] Tor can't help you if you use it wrong! Learn how to
be safe at https://www.torproject.org/download/download#warning
May 31 19:20:23.242 [notice] Read configuration file "/home/name/tor-9052/torrc".
May 31 19:20:23.249 [notice] Opening Socks listener on 127.0.0.1:9052
May 31 19:20:23.000 [notice] Parsing GEOIP IPv4 file /home/name/tor-9052/geoip.
May 31 19:20:23.000 [notice] Parsing GEOIP IPv6 file /home/name/tor-9052/geoip6.
May 31 19:20:24.000 [notice] Bootstrapped 0%: Starting
May 31 19:20:28.000 [notice] Bootstrapped 5%: Connecting to directory server
May 31 19:20:28.000 [notice] Bootstrapped 80%: Connecting to the Tor network
May 31 19:20:28.000 [notice] Bootstrapped 85%: Finishing handshake with first hop
May 31 19:20:30.000 [notice] Bootstrapped 90%: Establishing a Tor circuit
May 31 19:20:32.000 [notice] Tor has successfully opened a circuit. Looks like
client functionality is working.
May 31 19:20:32.000 [notice] Bootstrapped 100%: Done
```

Congratulations, when the additional tor routers are not being used they sleep and barely use a whisper of system resources.

If multiple running applications are all listening and reacting to the same proxy client port then you'll experience connectivity and stability problems in those applications. The same logic works with I2P concurrent applications regarding their active listening client ports. Many Regular Retroshare Node users now are running a proxy client Tor and I2P ports. If you run concurrent I2P applications make certain each is using a different listening, client socks proxy port. Creating an additional client Socks 4/4a/5 I2P Port is quite easy and takes you only a moment.

To create a new Socks 4/4a/5 I2P Proxy Tunnel Port

Start I2P
```
$ i2prouter start
```
Open your browser to http://127.0.0.1:7657/i2ptunnelmgr
At the bottom of the I2P Router Console webui page toggle 'New client tunnel' from Standard to Socks 4/4a/5 and select the 'Create' button.
New proxy settings
Enter New Tunnel Name, example Retroshare Proxy Client
Enter Your New I2P Access Point:Port
Check X on the Auto Start box
and select 'Save' at the bottom of the webui page.

Now back at the I2P Router Console page, select each I2P default tunnel that you do not intend to use and select the 'Stop' button on each. If you are likely to never use those additional I2P default tunnels then select each I2P tunnel name and in the Hidden Services Manager editor, uncheck its auto start box. Repeat this on each default start-up I2P tunnel you do not intend to use. You can always simply change them back later if you wish.

Each change is saved automatically so you can now close your browser if you do not need it for other purposes at this point.