

User Manual : Neosoft NILRT WireGuard Tools

Version 1.0

	Author	Auditor	Recipient
Name	Dany Baron		
Company	Neosoft Technologies	Neosoft Technologies	

Neosoft Technologies Inc.
4800 rue de Rouen, suite 230
Montréal, QC H1V 3T4

Toll-free: 1-866-636-7381
Phone: (418) 948-8324
Fax: (418) 948-8325
<https://www.neosoft.ca>

HISTORY

Version	Date	Author	Modification
1.0	2024-06-19	Dany Baron	Initial release

SUMMARY

1	Introduction	5
2	System Requirements	5
3	Licensing	5
4	Installation Instructions	6
4.1	Obtain the package	6
4.2	Install the package	6
4.2.1	If you have an .ipk file	6
4.2.2	If you have access to the Neosoft package repository	6
5	Using the software	7
5.1	Configuring WireGuard	7
5.1.1	Start the Configuration Tool	7
5.1.2	Interactive Configuration.....	7
5.1.3	Important information about Allowed IPs	9
5.1.4	Important information about the Endpoint URL	10
5.1.5	Important information about the Persistent Keepalive	10
5.2	Viewing current configuration	10
5.2.1	Show the current configuration for a WireGuard interface....	10
5.3	Starting and stopping WireGuard interfaces	11
5.3.1	Starting a WireGuard interface.....	11
5.3.2	Stopping a WireGuard interface	11
5.4	Starting a WireGuard interface at system startup	11
5.4.1	Enabling a WireGuard interface on system startup.....	11
5.4.2	Disabling the automatic startup of a WireGuard interface	11
6	Troubleshooting	12

FIGURES

Figure 1.	Neosoft license directory	5
Figure 2.	Installing neo-wg-tools from file.....	6
Figure 3.	Interface configuration example	7
Figure 4.	Generating a peer	8
Figure 5.	Peer information after configuration.....	9

1 Introduction

Neosoft NILRT WireGuard Tools (neo-wg-tools) is a software package designed for managing WireGuard interface configurations on National Instruments Linux Real-Time (NILRT) systems. This tool provides a user-friendly, command-line interface for configuring WireGuard interfaces to enhance the security of your NILRT devices.

2 System Requirements

- Target Hardware: National Instruments CompactRIO and PXI products
- Operating System: NI Linux Real-Time (NILRT) 2023 Q1 or later
- Prerequisites:
 - Root access to the target device
 - SSH client (e.g., PuTTY) for remote access

3 Licensing

The neo-wg-tools package requires a license from Neosoft Technologies specific to the unit's serial number.

1. Obtain a License:
Contact Neosoft Technologies to obtain a license file (<package>.lic) for your target device. You may contact Neosoft Technologies using the contact information at the start of this document. Be sure to provide the serial number of the target on which you wish to install the package.
2. Install the License:
Transfer the license file to the /etc/neosoft/licenses directory on the target device using a SFTP client. You may create the directory if it does not exist.

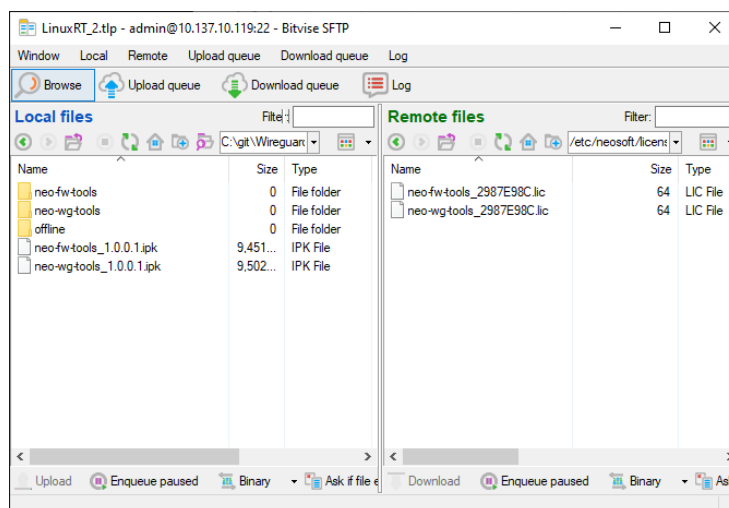


Figure 1. Neosoft license directory

4 Installation Instructions

4.1 Obtain the package

You may add the Neosoft package repository if you were provided an URL or obtain the package file directly from your Neosoft representative.

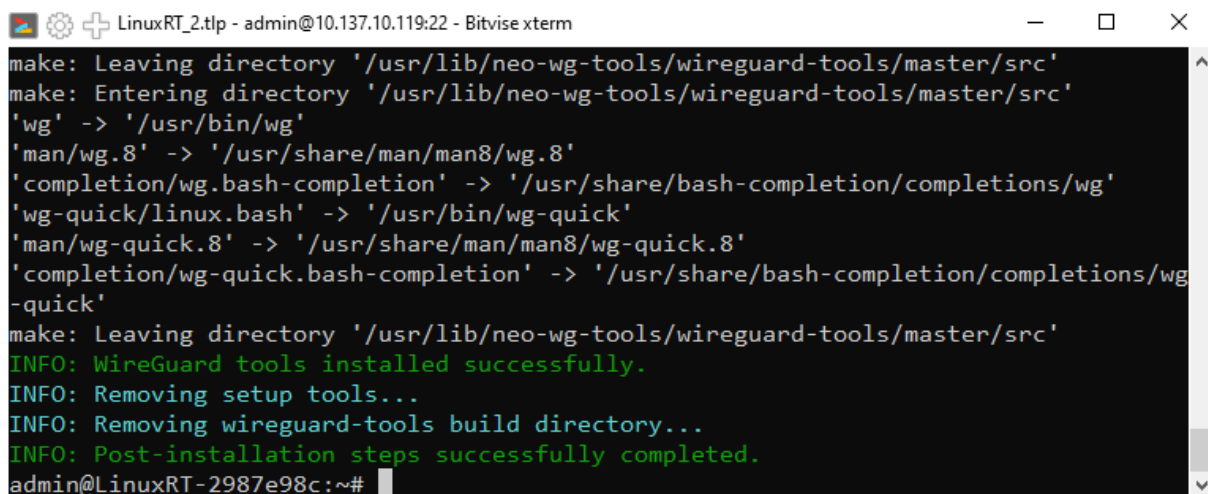
If you obtained the package file directly, use a SFTP client to transfer the package file (<package>_<version>.ipk) to the target device.

4.2 Install the package

4.2.1 If you have an .ipk file

Open an SSH terminal (e.g., PuTTY) and connect to the target device. Execute the following command to install the package (replace <package.ipk> with the actual name of the file you uploaded to the device):

```
opkg install <package.ipk>
```



```
LinuxRT_2.tlp - admin@10.137.10.119:22 - Bitvise xterm
make: Leaving directory '/usr/lib/neo-wg-tools/wireguard-tools/master/src'
make: Entering directory '/usr/lib/neo-wg-tools/wireguard-tools/master/src'
'wg' -> '/usr/bin/wg'
'man/wg.8' -> '/usr/share/man/man8/wg.8'
'completion/wg.bash-completion' -> '/usr/share/bash-completion/completions/wg'
'wg-quick/linux.bash' -> '/usr/bin/wg-quick'
'man/wg-quick.8' -> '/usr/share/man/man8/wg-quick.8'
'completion/wg-quick.bash-completion' -> '/usr/share/bash-completion/completions/wg-quick'
make: Leaving directory '/usr/lib/neo-wg-tools/wireguard-tools/master/src'
INFO: WireGuard tools installed successfully.
INFO: Removing setup tools...
INFO: Removing wireguard-tools build directory...
INFO: Post-installation steps successfully completed.
admin@LinuxRT-2987e98c:~#
```

Figure 2. Installing neo-wg-tools from file.

4.2.2 If you have access to the Neosoft package repository

Open an SSH terminal (e.g., PuTTY) and connect to the target device. Execute the following command to download and install the package:

```
opkg install neo-fw-tools
```

5 Using the software

5.1 Configuring WireGuard

The wireguard-config utility is designed to simplify the process of setting up and managing WireGuard VPN interfaces on your NILRT device.

5.1.1 Start the Configuration Tool

Open an SSH terminal and connect to the target device. Execute the following command with root privileges to start the configuration tool:

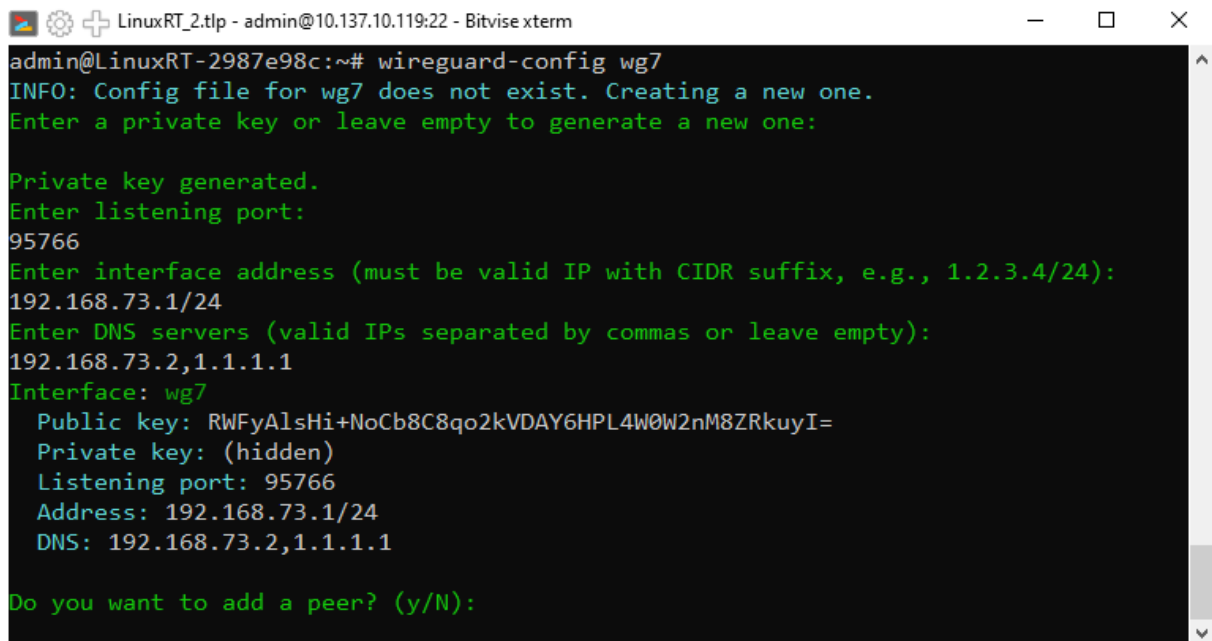
```
sudo wireguard-config <interface_name>
```

If <interface_name> is not provided, the utility defaults to creating/editing interface "wg0".

5.1.2 Interactive Configuration

Follow the interactive prompts to configure your WireGuard interface. You will be guided through the following steps:

- **Key Generation:** Generate or specify a private key for the interface.
- **Interface Parameters:** Specify the IP, network, listening port and optionally DNS servers for the virtual interface.
- **Peer Configuration:** Add and configure peers, including setting up pre-shared keys, allowed IPs, endpoint, and persistent keepalive settings.



```
LinuxRT_2.tlp - admin@10.137.10.119:22 - Bitvise xterm
admin@LinuxRT-2987e98c:~# wireguard-config wg7
INFO: Config file for wg7 does not exist. Creating a new one.
Enter a private key or leave empty to generate a new one:

Private key generated.
Enter listening port:
95766
Enter interface address (must be valid IP with CIDR suffix, e.g., 1.2.3.4/24):
192.168.73.1/24
Enter DNS servers (valid IPs separated by commas or leave empty):
192.168.73.2,1.1.1.1
Interface: wg7
  Public key: RWFyAlSHi+NoCb8C8qo2kVDAY6HPL4W0W2nM8ZRkuyI=
  Private key: (hidden)
  Listening port: 95766
  Address: 192.168.73.1/24
  DNS: 192.168.73.2,1.1.1.1

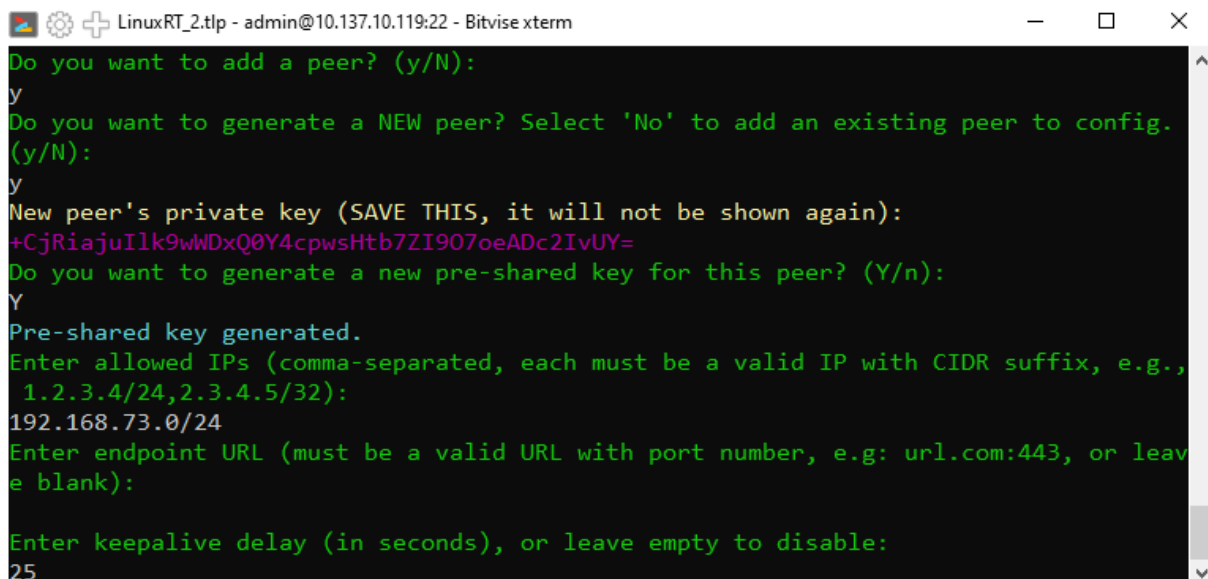
Do you want to add a peer? (y/N):
```

Figure 3. Interface configuration example

The first step is to provide or generate a key for the interface you wish to create. It is recommended to generate it locally if possible, to protect the private key, as it will be hidden during the entire process.

The second step is to configure the interface as shown in Figure 3. You will need to provide an IP address for the interface itself with the CIDR suffix of the virtual network on which it will operate. This means that other peers of the interface must be part of the same subnet. For example, if you use 192.168.10.1/24 (subnet mask 255.255.255.0), the peers must use addresses within the 192.168.10.1-192.168.10.254 range, excluding the current interface IP, in order to communicate.

The DNS servers, if provided, are the ones **for the virtual network**. If DNS servers are not used within the virtual network, it may be ignored. If using the virtual network as an access to the Internet, you may add a public DNS here, but it is recommended to use a local DNS relay instead. Note that routing to physical networks is not done automatically with WireGuard (as it is a "serverless" infrastructure) so if using the virtual network to reach a remote physical network, more advanced routing configuration will be required outside of the WireGuard configuration.

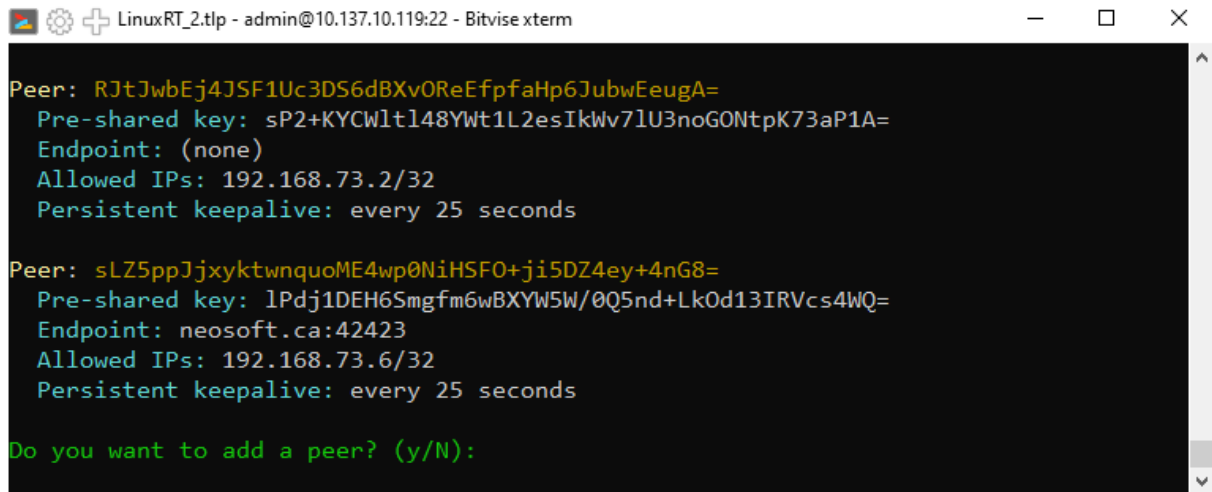


```
LinuxRT_2.tlp - admin@10.137.10.119:22 - Bitvise xterm
Do you want to add a peer? (y/N):
y
Do you want to generate a NEW peer? Select 'No' to add an existing peer to config.
(y/N):
y
New peer's private key (SAVE THIS, it will not be shown again):
+CjRiajuIlk9wWDxQ0Y4cpwsHtb7ZI907oeADc2IvUY=
Do you want to generate a new pre-shared key for this peer? (Y/n):
Y
Pre-shared key generated.
Enter allowed IPs (comma-separated, each must be a valid IP with CIDR suffix, e.g.,
1.2.3.4/24,2.3.4.5/32):
192.168.73.0/24
Enter endpoint URL (must be a valid URL with port number, e.g: url.com:443, or leave
blank):
Enter keepalive delay (in seconds), or leave empty to disable:
25
```

Figure 4. Generating a peer

If you wish to add peers at this point, you may either generate a new peer with its own private and public key or add an already existing peer that you created elsewhere.

When generating a peer, you **MUST** save the generated private key as it will not be shown later or stored. This key will be used to set up the interface on the peer. If a pre-shared key has been generated, it will be shown later along with the public key.



```
LinuxRT_2.tlp - admin@10.137.10.119:22 - Bitvise xterm

Peer: RJtJwbEj4J3SF1Uc3DS6dBXv0ReEfpaHp6JubwEeugA=
  Pre-shared key: sP2+KYCWlt148Ywt1L2esIkWv7lU3noGONtpK73aP1A=
  Endpoint: (none)
  Allowed IPs: 192.168.73.2/32
  Persistent keepalive: every 25 seconds

Peer: sLZ5ppJjxyktnquoME4wp0NiHSFO+ji5DZ4ey+4nG8=
  Pre-shared key: lPdjl1DEH6Smgfm6wBXYW5W/0Q5nd+LkOd13IRVcs4WQ=
  Endpoint: neosoft.ca:42423
  Allowed IPs: 192.168.73.6/32
  Persistent keepalive: every 25 seconds

Do you want to add a peer? (y/N):
```

Figure 5. Peer information after configuration

When adding an existing peer, you will need to add the public key and pre-shared key (if used).

In both cases, you will need to specify the allowed IPs, the endpoint URL and the keepalive delay.

Once you are done adding peers, enter "N" or press the enter key to generate the interface configuration.

Executing the command again on an already-configured interface will allow you to add peers to it. This means that you can configure the interface first and then add the peers later if you wish. This is useful if you have multiple devices and want to configure all of them first before adding the peers between them.

5.1.3 Important information about Allowed IPs

The allowed IPs parameter is a list of IPs that can be used on this interface, with packets from other IPs coming from this peer being rejected.

However, it is not its only purpose. **This parameter is also used to configure the local outbound routes used to communicate with the peer.** This means that IPs configured here is how the system will know which IPs is owned by which peer. If this is not configured properly, the local system will not be able to communicate with the peer.

For example, if the remote peer is a single machine and not a whole network (which is the most common use case), the only allowed IP for the peer should be its interface IP with the /32 CIDR suffix (e.g. 192.168.73.3/32). This will cause a static route to be created on the local machine when the interface is activated, causing packets going to that IP to be sent through the WireGuard interface and then routed to the right peer. Otherwise, the local system will not have any way to route the packets to the destination.

5.1.4 Important information about the Endpoint URL

The endpoint URL that is configured for a peer is the address and port that will **actively** be contacted by the local WireGuard interface. It is not required, as some peers may be passive, meaning that they will wait to be contacted by the peer rather than actively connect to it (like a server).

For any set of 2 peers connecting to each other, at least one must have an endpoint configured. Since otherwise all peers are passive and no connection will be made.

For example, you may want to have a peer acting as a "server" to the other peers. The "server" peer will have no endpoint configured, but all the "client" peers have the "server" endpoint configured in their own peers, allowing them to contact the server to establish a connection. The "server" will only have to accept the inbound connections.

Another possibility is that you may have a peer that is behind a NAT rather than exposed to the wider network (for example the internet). The peer that is behind the NAT can create an outbound connection to the endpoint peer (that is exposed) allowing for the connection to be made through the NAT without exposing any ports on the "client" side.

The endpoint may be a hostname or an IP address, with the port being the one of the *WireGuard interface* being contacted.

5.1.5 Important information about the Persistent Keepalive

The Persistent Keepalive setting is an optional setting allowing for the WireGuard interface to send keepalive packets to that peer at the specified interval. The main usage of this setting is in the case (discussed in section 5.1.4) where a peer is behind a NAT in order to keep the connection alive in the router and prevent packet loss when there is no more data being transmitted between the peers for a long time. **Keep in mind that the WireGuard protocol, by default, does not send packet continuously between peers, unlike more traditional VPN solutions.** This makes the WireGuard protocol very efficient, but the keepalive will be required in these cases.

A safe value for Persistent Keepalive, if used, is 25 seconds.

5.2 Viewing current configuration

The **wireguard-show** utility displays the current configuration of WireGuard interfaces, providing a detailed view of interface and peer settings.

5.2.1 Show the current configuration for a WireGuard interface

To display the configuration of a WireGuard interface, use the following command:

```
sudo wireguard-show <interface_name>
```

If the interface name is not provided, the utility defaults to displaying the configuration for wg0.

The utility will provide a detailed output, including:

Interface Details: Public keys, listening port, IP address, and DNS settings.

Peer Details: Public keys, pre-shared keys, endpoints, allowed IPs, and keepalive settings. The private key is hidden to maintain its confidentiality.

5.3 Starting and stopping WireGuard interfaces

The **wireguard-start** and **wireguard-stop** utilities allow you to manually start and stop WireGuard interfaces on your NILRT device.

5.3.1 Starting a WireGuard interface

To manually start a WireGuard interface, use the following command:

```
sudo wireguard-start <interface_name>
```

If the interface name is not provided, the utility defaults to starting wg0.

5.3.2 Stopping a WireGuard interface

To manually stop a WireGuard interface, use the following command:

```
sudo wireguard-stop <interface_name>
```

If the interface name is not provided, the utility defaults to stopping wg0.

5.4 Starting a WireGuard interface at system startup

The **wireguard-enable-startup** and **wireguard-disable-startup** utilities manage the automatic startup of WireGuard interfaces on your NILRT device.

5.4.1 Enabling a WireGuard interface on system startup

To enable a WireGuard interface to start automatically on system boot, use the following command:

```
sudo wireguard-enable-startup <interface_name>
```

If the interface name is not provided, the utility defaults to enabling startup for wg0.

5.4.2 Disabling the automatic startup of a WireGuard interface

To disable a WireGuard interface from starting automatically on system boot, use the following command:

```
sudo wireguard-disable-startup <interface_name>
```

If the interface name is not provided, the utility defaults to disabling startup for wg0.

6 Troubleshooting

- **Permission Denied:** Ensure you are running any of the utilities with root privileges, as it is required to modify the network configuration.
- **License validation issues:** Verify the license file is placed in the correct directory (/etc/neosoft/licenses) and corresponds to the target device's serial number.