

# Installation on Windows

*Qosium has an easy-to-use installer available for Windows. The installation process is very straightforward and is completed typically within a minute.*

## Table of Contents

1. Installing a Packet Capture Library on Windows .....	3
1.1. Npcap .....	3
1.2. Win10Pcap .....	4
1.3. WinPcap .....	4
1.4. Parameterization: Timestamping Mode .....	5
2. Clock Synchronization in Windows .....	5
2.1. Introduction .....	5
2.2. NTP Client: Automatic Configuration .....	6
2.3. NTP Client: Manual Configuration .....	6
2.4. NTP Server Configuration .....	7
3. Glossary .....	8

Qosium's Windows installer contains typically the Qosium *GP* including **Qosium Probe**, **Qosium Scope**, **Qosium Scopemon**, and **Qosium Scope Lite**. A separate installation package only for Qosium Probe can be provided as well if needed.

In some cases, the customer has their own installation routines including software packaging. Thus, if needed, Kaitotek can also deliver Qosium Probe's pure binary files without the installation package.

Follow these steps to install Qosium:

1. Install [a packet capture library](#), if not already installed
  2. [Sign in](#) to your account
  3. Download the installer from [your downloads page](#)
  4. Launch the installer: an installation wizard guides you through the installation process.
- At the end of the installation, the wizard asks for permission to set up the timing related parameters. If the clock synchronization settings of the environment are already handled, you can ignore this, but if not, it is recommended to allow Qosium to perform the setup.
  - If you chose to ignore the timing settings, take a look at, still, the [packet capturer timestamping parameterization](#), since that needs some attention.

After completing these steps, Qosium Probe has been successfully installed to your system.

## 1. Installing a Packet Capture Library on Windows

The typical installation package of Qosium does not contain a packet capturing library. If a such library has not been installed in the system earlier, it has to be installed manually. There is also some parameterization to consider.

In Windows, there are at least three supported alternatives for packet capturing, which are discussed in the following sections.

Probe 1.9.2.0

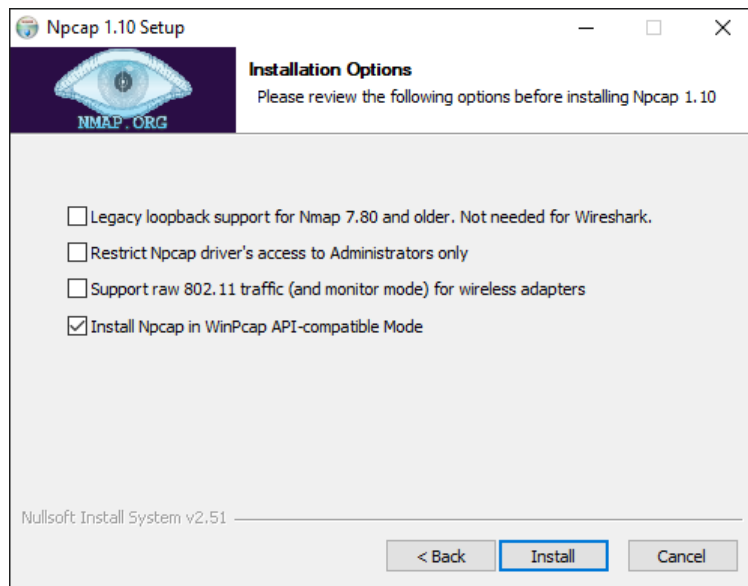
Only Npcap is supported from this version forward.

### 1.1. Npcap

Npcap is the recommended packet capturing library to use with Qosium. It is originally based on WinPcap, but it is under continuous development, unlike WinPcap. Npcap uses *NDIS* 6.x, so it should support also those *NIC*s that WinPcap no longer does. Please bear in mind that in contrast to WinPcap, Npcap's free use is not unlimited. Thus, please check the license conditions of Npcap if you are considering using it.

[Download Npcap](#)

If you choose Npcap, the older versions of Qosium (Probe version before 1.9.2.0 and Scope version before 2.8.1.5) require it to be installed in WinPcap-compatible mode. See below what needs to be checked in the installation wizard. The other options shown in the figure are up to the needs – they are neither limited nor required by Qosium.



If you need to make modifications to the driver's parameters, restart the driver, e.g., from the command prompt (with superuser rights) to enable the changes.

To shut down:

```
net stop npcap
```

To start:

```
net start npcap
```

## 1.2. Win10Pcap

Another alternative for a packet capturing driver for Windows is Win10Pcap. Like Npcap, this is also based on WinPcap but has the support for the NDIS 6.x driver model. Win10Pcap works directly with the older versions of Qosium. Win10Pcap is fully free to use since it is under the GPLv2 license. It is, however, likely that the driver is no longer developed: the newest version is from Oct. 8, 2015. In any case, it is still newer than the original WinPcap and seems to work.

One technical downside is that it is not confirmed whether the different timestamp modes are supported or not in Win10Pcap. At least, with tests carried out, we haven't yet been successful in modifying the timestamp mode.

[Download Win10Pcap](#)

## 1.3. WinPcap

WinPcap (version 4.1.3) is one option for the older versions of Qosium. Despite the fact that it has not been updated for years, and uses the nowadays deprecated NDIS 5.x, it still works with most of the system configurations at least in Windows 10. Thus, if your NIC works with WinPcap, there is no imminent reason to stop using it.

[Download WinPcap](#)

## 1.4. Parameterization: Timestamping Mode

Npcap and WinPcap support several timestamp modes. With the default mode 0, the absolute accuracy is “a one-shot accuracy”, since system performance counters are used for providing high timestamp resolution, but the absolute time is not synchronized. This means that once the driver is turned on, the absolute time is once got from the system clock, but after this, system performance counters will be used for updating the time. Hence, the absolute time begins to drift, and the accuracy of Qosium’s delay measurement gets worse as a function of measurement time. However, the resolution is high – even better than in Linux – leading to highly accurate jitter calculation.

If you are interested in one-way delay measurements, it is recommended to change the timestamp mode. One option supported by all known versions of Npcap and WinPcap is 2, in which case, the system clock is used for timestamping packets. This relieves the one-shot accuracy problem but has a downside that it makes the accuracy of jitter calculation worse. This is thanks to the Windows’ system clock resolution that is typically only 1 ms. In fact, the default Windows’ system clock resolution is as bad as 15.6 ms, but when Qosium is in use, it increases the resolution to 1 ms. For most of the practical applications this 1 ms resolution is already enough, but of course, it is very far from the  $\mu$ s-level resolution that can be reached with timestamp mode 0.

The newer versions of Npcap support also timestamp mode 4. In this, the system clock is used for absolute timing, but system performance counters are used to calculate the accurate time between the sparse clock ticks. If you can use a new Npcap, this is the recommended mode, since it provides highly accurate timestamps without losing the absolute timing synchronization.

If you allow, Qosium’s installation wizard detects automatically the installed Pcap version and selects timestamp mode 4 when available and mode 2 otherwise. If not, here are the instructions on how to do it manually. The timestamping mode can be changed in the registry. Prior to this, the packet capturing driver must be stopped.

1. Stop the measurement first (and all SW using Pcap, such as Wireshark in addition to Qosium)
2. Open the command prompt and stop the packet capturing driver as explained earlier in the context of Npcap
3. Open *Registry Editor* and navigate to  
WinPcap: `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NPF`  
Npcap (below 0.9985): `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\npf\Parameters`  
Npcap (version 0.9985 and higher):  
`HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\npcap\Parameters`
4. Locate the parameter **TimestampMode**. If it does not exist, create it (REG\_DWORD).
5. Change/set the mode to the desired value
6. Start the packet capturing driver

### Probe 1.9.2.0

From this version forward, manual timestamp mode parameterization is no longer needed. Qosium Probe is itself capable of selecting always the best available mode.

## 2. Clock Synchronization in Windows

Windows has an inbuilt NTP client, so there is no need to install another NTP software. However, the standard settings in the NTP client are only meant for rough system timing. These settings can be modified in order to get better results.

### 2.1. Introduction

System clock synchronization in Windows is carried out by its own *NTP* client. In the Services-list it can be

found with the name Windows Time and as a driver, it is called *W32Time*. The default parameters of the Windows NTP client are not at their best for accurate measurement purposes. By tweaking the parameters, typical NTP absolute clock accuracy of some milliseconds can be reached. This requires a high-quality network connection with low jitter to the NTP server. If synchronization is carried out over wireless, such as WiFi, the accuracy will be worse.

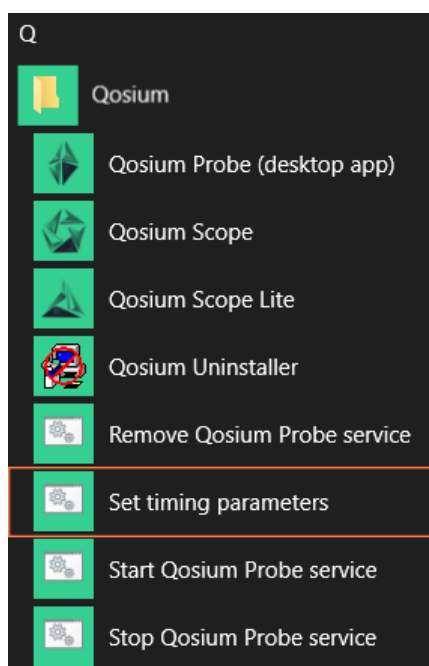
With Qosium's installation package, a set of Windows NTP parameters is also given. By using these, typically a reasonable accuracy for measurement is reached. Those are discussed in the next section.

If the goal is to perform highly accurate delay measurements, other clock synchronization methods are recommended. By using an external *GNSS* clock with Qosium, it is possible to reach about 50  $\mu$ s absolute timing accuracy for delay measurements with 1  $\mu$ s jitter resolution (at its best). This, however, requires a custom synchronization driver, which is currently at an experimental state. If you are interested in this, please contact Kaitotek's support.

The *W32Time* driver in the newer Windows 10 based systems supports also *PTP*. This leads to much better accuracy when compared to NTP. If you are interested in this, contact Kaitotek's support for instructions.

## 2.2. NTP Client: Automatic Configuration

Qosium's installation wizard will ask at the end permission to tweak the Windows' NTP client settings. If you did not allow this, it can be done later from the shortcut. Remember, you need to run the script with superuser rights.



## 2.3. NTP Client: Manual Configuration

Windows NTP client parameters can also be set manually. Qosium's installation package contains a registry file that can be taken into use easily. It is found at the Qosium's installation folder with name `win_timing_conf_<code>.reg`, where the `<code>` part is dependent on the distribution. Before activating the new settings, you can also edit them by a text editor. Alternatively, you can set the values directly with Registry Editor.

To activate the parameters from the file:

1. Open Windows Command Prompt with superuser (system administrator) rights

2. Stop the W32Time service by typing: `net stop w32time`
3. Go to the Qosium's installation folder by using, e.g., File Explorer
4. Double-click the file `win_timing_conf_<code>.reg` and allow it to be entered to Registry
5. At the Command Prompt, start the W32Time service by typing: `net start w32time`

Please note that it might take some minutes before the synchronization takes effect.



There might also be other timing clients installed to the system. In this case, in order to use the Windows' own NTP client, find the other clients and shut them down.

## 2.4. NTP Server Configuration

Windows NTP can also be used in server mode to distribute time to NTP clients. The NTP server is often, however, turned off by default. To check and turn on the NTP server, do the following:

1. Open Windows Registry Editor tool
2. Go to:  
Computer\HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\W32Time\TimeProviders\NtpServer
3. Check that the value **Enabled** is set to 1. If not, change it.
4. Close Registry Editor

If you made changes, you need to restart the Windows Time service. This can be done in many ways, e.g., by using Command Prompt:

1. Open Windows Command Prompt with superuser rights
2. Stop the W32Time service by typing: `net stop w32time`
3. Start the W32Time service by typing: `net start w32time`

## 3. Glossary

### General Package

*Qosium General Package contains the basic Qosium components required for measurements and monitoring.*

### Network Driver Interface Specification

*NDIS on Windowsin ohjelmointirajapinta verkkokortelle*

[Wikipedia article on Network Driver Interface Specification](#)

### Network Interface Card

*A piece of hardware which offers a device a networking interface.*

### Network Time Protocol

*A very common protocol for synchronizing the clocks of devices across a network.*

### Global Navigation Satellite System

*A general term under which all the different global satellite navigation systems (e.g., GPS, GLONASS, Galileo, BDS) fall.*

### Precision Time Protocol

*A protocol for synchronizing the clocks of devices across a network. The reached synchronization accuracy is typically considerably better than with NTP.*