**Last Update: 4/3/19**

RS5 Validation Guide

**Summary:** This document provides implementation steps to implement and test the Precision Time Protocol client on Windows 10 and Server 2019.

This guide should be accompanied by the Software Timestamping and Time Accuracy Validation guides found here:

* Software Timestamping: [Link](https://github.com/Microsoft/W32Time/tree/master/Timestamping/docs)
* High Accuracy Configuration: [Link](http://docs.microsoft.com/en-us/windows-server/networking/windows-time-service/configuring-systems-for-high-accuracy)

Table of Contents

[Overview 3](#_Toc5216002)

[Description 3](#_Toc5216003)

[Feedback 4](#_Toc5216004)

[Test Activities 4](#_Toc5216005)

[Activities 4](#_Toc5216006)

[Activity 1: Configure the System for High Accuracy 4](#_Toc5216007)

[Activity 2: Configure the PTP Slave 4](#_Toc5216008)

[Activity 3: Configure the Firewall 5](#_Toc5216009)

[Activity 4: Verify PTP Provider Configuration 5](#_Toc5216010)

[Troubleshooting 6](#_Toc5216011)

[Recommended Steps 6](#_Toc5216012)

[PTPd Grandmaster 8](#_Toc5216013)

[LinuxPTP (ptp4l) Grandmaster 9](#_Toc5216014)

# Overview

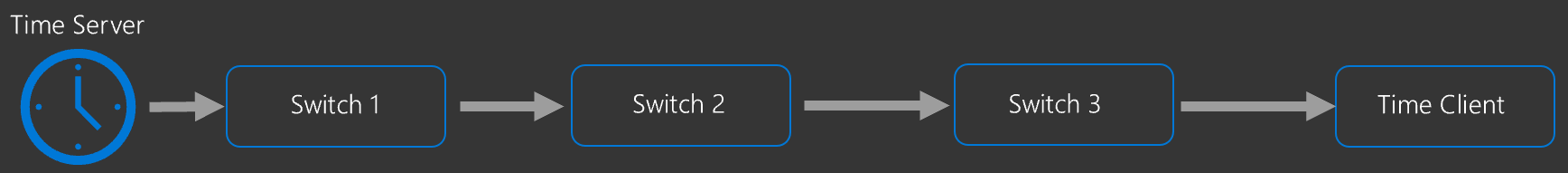
Many sectors require accurate time. The financial sector requires accurate time so that transactions are properly timestamped to 50ms, 1ms or even 100µs accuracy. Windows Server 2019 and the corresponding Windows 10 release continues to improve Windows’ accuracy by leveraging a new time synchronization client protocol, Precision Time Protocol.

# Description

Think back to the last thunderstorm you saw  – Did you see lightning and hear thunder at the same time?  Unless you’re very close to the storm, you’ll likely detect an audible delay after you’ve seen the lightning.  How much of an audible delay are you experiencing?  The delay is not based strictly on the speed of sound and your distance from the storm.  It's also affected by buildings or other influences that introduce additional acoustic delay.  If you want to know just how close to the storm you are, you'd have to consider all the influences.

[](https://msdnshared.blob.core.windows.net/media/2018/06/tstorm.png)

Likewise, there is delay (latency) introduced in the timing packets being passed from the time server across the network.  If that delay is not accounted for, or if it is not symmetric (equal in both directions – to and from the client), then it becomes increasingly difficult for the client to properly apply the time stamp sent from the time server.

[](https://msdnshared.blob.core.windows.net/media/2018/06/TIme2Client.png)

Network Time Protocol (NTP) has long been the primary time synchronization method for Windows but unfortunately, NTP does not have a solution to this problem; NTP assumes that the round-trip delay introduced by the network is symmetric.

Precision Time Protocol ([IEEE 1588v2](https://standards.ieee.org/findstds/standard/1588-2008.html)) enables network devices to add the latency introduced by each network device into the timing measurements thereby providing a far more accurate time sample to the endpoint (Windows Server 2019 or Windows 10, host or virtual machine).

Precision Time Protocol is not for everyone; due to the network configuration requirements, NTP will continue to be supported (and default) protocol.  However, for customers with the highest of accuracy requirements, you can drive towards even higher accuracy systems using our inbox PTP Client in Windows Server 2019.

Operating System Requirements

Software timestamping is currently supported only on Windows Server 2019 and Windows 10 (v1809). This feature is available on current Insider builds.

Protocol Requirements

The Windows PTP Provider supports IEEE 1588v2 with the default profile only. For additional troubleshooting support, please see the [Troubleshooting](#_Troubleshooting) section.

# Feedback

Please submit an issue at: <https://aka.ms/W32Time/issues>

# Test Activities

Below are the activities included to validate this feature:

1. Configure the System for High Accuracy
2. Configure the PTP Slave
3. Configure the Firewall
4. Verify PTP Provider Configuration

# Activities

## Activity 1: Configure the System for High Accuracy

PTP is intended for the highest accuracy time. As such you should also configure your system for high accuracy prior to proceeding. The high accuracy settings and configuration can be found here: [Link](https://docs.microsoft.com/en-us/windows-server/networking/windows-time-service/configuring-systems-for-high-accuracy?tabs=MinPollInterval)

## Activity 2: Configure the PTP Slave

This activity will help you configure the Windows PTP Slave to communicate with a third-party PTP Time server.

First, configure the following registry keys to configure PTP and disable the other client providers (NTP, VMIC) to avoid confusion.

**Note**: Modify the following entry 10.10.10.10 10.10.10.11 as a space-separated list of your PTP grandmasters

**Note**: To use Multicast (off by default) change the EnableMulticastRx to 1

|  |  |  |
| --- | --- | --- |
| **Root Key** | HKLM\SYSTEM\CurrentControlSet\Services\W32Time\TimeProviders\PtpClient | |
| **Type** | **Name** | **Value** |
| REG\_SZ | PtpMasters | 10.10.10.10 10.10.10.11 |
| REG\_DWORD | Enabled | 1 |
| REG\_DWORD | InputProvider | 1 |
| REG\_SZ | DllName | "c:\windows\system32\ptpprov.dll" |
| REG\_DWORD | DelayPollInterval | 0x3e80 |
| REG\_DWORD | AnnounceInterval | 0x0fa0 |
| REG\_DWORD | EnableMulticastRx | 0 |
| **Root Key** | HKLM\SYSTEM\CurrentControlSet\Services\W32Time\TimeProviders\NtpClient | |
| REG\_DWORD | Enabled | 0 |
| **Root Key** | HKLM\SYSTEM\CurrentControlSet\Services\W32Time\TimeProviders\VMICTimeProvider | |
| REG\_DWORD | Enabled | 0 |

Next, restart the Windows Time Service.

**Note:** See [this link](https://github.com/Microsoft/W32Time/tree/master/Precision%20Time%20Protocol/Windows%20Configuration%20Helpers) for some simple automation help of these settings.

## Activity 3: Configure the Firewall

Create the following Firewall rules to allow the PTP Client to communicate with the time server.

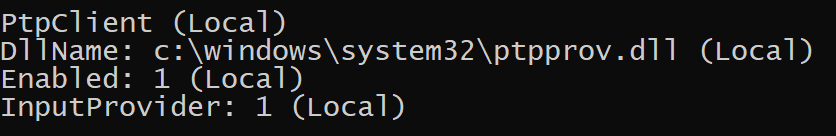
|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Direction** | **Protocol** | **LocalPort** |
| PTP Event In | In | UDP | 319 |
| PTP Event Out | Out | UDP | 319 |
| PTP General In | In | UDP | 320 |
| PTP General Out | Out | UDP | 320 |

**Note:** See [this link](https://github.com/Microsoft/W32Time/tree/master/Precision%20Time%20Protocol/Windows%20Configuration%20Helpers) for some simple automation help of these settings.

## Activity 4: Verify PTP Provider Configuration

1. Verify that the PTP Time Provider is loaded and enabled by running the command **w32tm /query /configuration**

Each loaded provider will have an entry under the **[TimeProviders]** section. The **Enabled** property will be 1 if it is enabled.



1. Now run the command **w32tm /resync**

**Note:** This command will not cause synchronize with PTP clients. Instead, it asks the PTP Provider to re-evaluate the grandmaster. Please be patient as this can take a few minutes. Review the event logs

1. Verify that the system is synchronizing with the PTP Source Server

**w32tm /query /status /verbose**

*Leap Indicator: 0(no warning)*

***Stratum: 3*** *(secondary reference - syncd by (S)NTP)*

*Precision: -23 (119.209ns per tick)*

*Root Delay: 0.0006638s*

*Root Dispersion: 0.0100020s*

*ReferenceId:* ***0x4D505450 (source IP:  77.80.84.80)*** *– This indicates Microsoft PTP Provider*

*Last Successful Sync Time: 5/1/2018 7:10:46 PM*

***Source: 10.11.12.1    – This indicates your specific PTP Server***

*Poll Interval: 6 (64s)*

*Phase Offset: 0.0368311s*

*ClockRate: 0.0156250s*

*State Machine: 1 (Hold)*

*Time Source Flags: 0 (None)*

*Server Role: 0 (None)*

***Last Sync Error: 0 (The command completed successfully.)    – This indicates your last sync was successful***

*Time since Last Good Sync Time: 2.8389207s*

In the output above, the Reference ID ***0x4D505450 (source IP:  77.80.84.80)*** indicates that the PTP Provider is in use. If you don’t see this, then a non-disabled PTP provider you may not have disabled the other providers.

**Note**: Why 0x4D505450? This is the ASCII Conversion for “MPTP” (Microsoft PTP)

# Troubleshooting

Please note: As Microsoft provides a PTP Client, troubleshooting various grandmaster implementations is out-of-scope for support however we will include known issues to assist in the configuration of the Windows Client.

Mention of any specific grandmaster should not be considered advertisement, recommendation, or any other type of endorsement of said grandmaster. Specifics here are only included to assist in troubleshooting.

## Recommended Steps

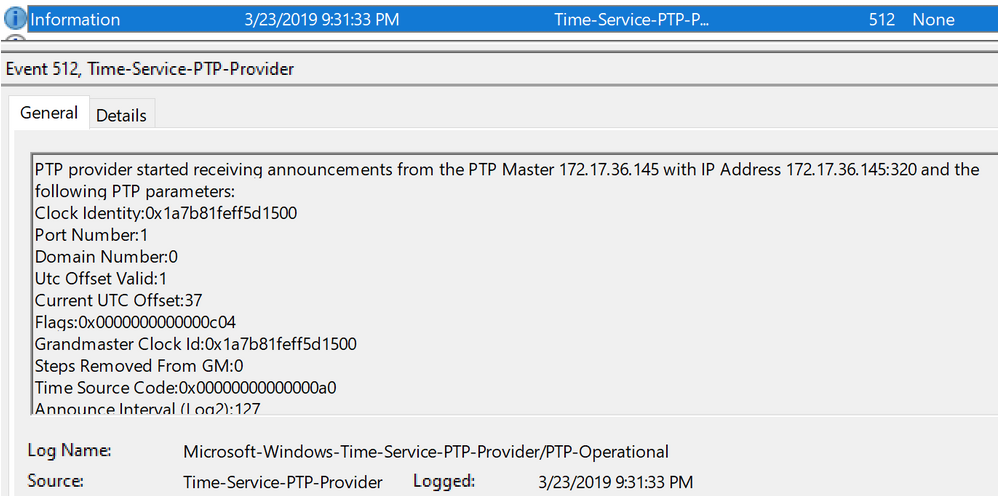
If you’re system is unable to receive time from the grandmaster, please verify:

1. You are receiving PTP Messages on the Windows client for UDP ports 319 and 320. You can use wireshark to filter only PTP traffic with the following filter:

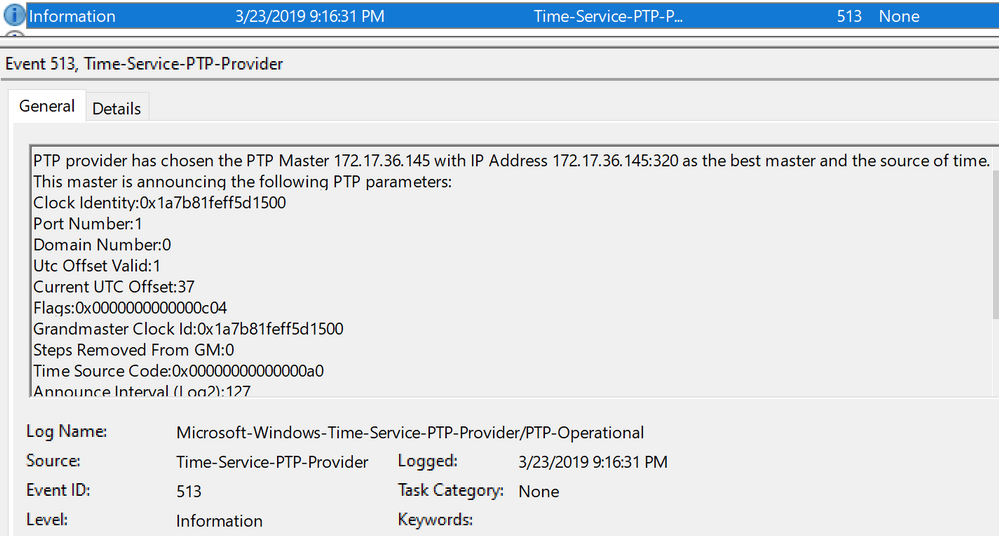
udp.port in {319 320}

1. You are seeing the event logs at **Event Viewer** > **Applications and Services** > **Microsoft**> **Windows**>**Time-Service-PTP-Provider** > **PTP-Operational**

**Event 512 Indicates that announce messages were received**

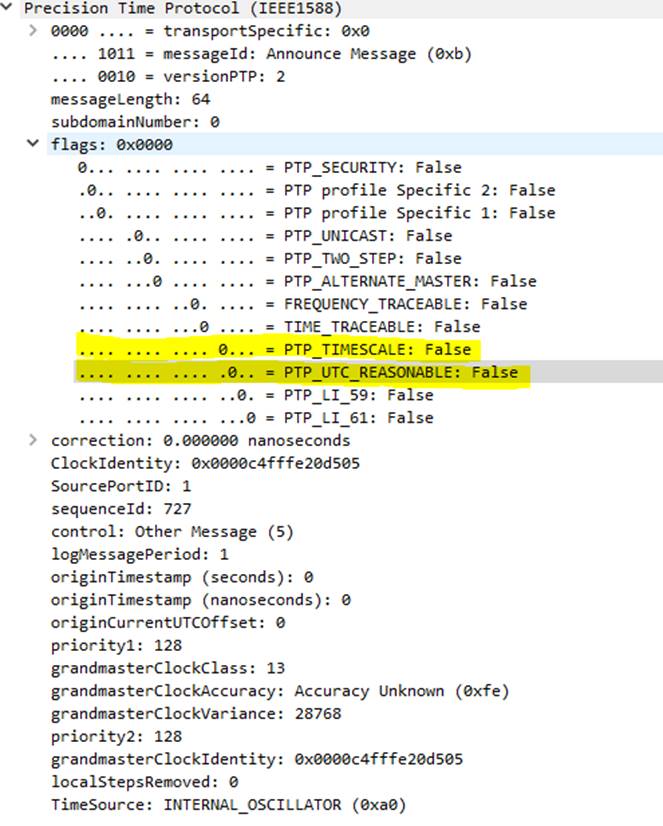
****

**Event 513 Indicates that a grandmaster has been selected**

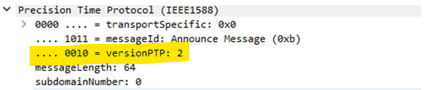
****

1. The Grandmaster properly advertises the **PTP\_TIMESCALE** and **PTP\_UTC\_REASONABLE** flags in the announce messages

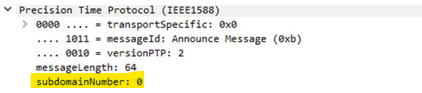
For security measures, the Windows PTP Client will only synchronize with a grandmaster if its **PTP\_TIMESCALE** and **PTP\_UTC\_REASONABLE** flags are non-zero.



1. The grandmaster is using PTP (IEEE 1588) version 2



1. The PTP domain number must be 0



## PTPd Grandmaster

Microsoft validated its PTP Slave with multiple grandmasters including [PTPd](https://github.com/ptpd/ptpd/commits/master). Please refer to [this Microsoft GitHub location](https://github.com/Microsoft/W32Time/tree/master/Precision%20Time%20Protocol/PTPd%20Configuration%20Examples) for example configurations with PTPd.

For a simple example of how to test this using our common configuration, please see this [blog and video](https://techcommunity.microsoft.com/t5/Networking-Blog/Windows-Subsystem-for-Linux-for-testing-Windows-10-PTP-Client/ba-p/389181)

## LinuxPTP (ptp4l) Grandmaster

According to various documentation sources ([including here](http://rpm.pbone.net/index.php3/stat/45/idpl/24646060/numer/5/nazwa/ptp4l.conf))

“In software and legacy time stamping modes it announces Arbitrary timescale mode, which is effectively UTC here, in hardware time stamping mode it announces use of PTP time scale.”

As previously noted, the Windows PTP Client requires the PTP\_TIMESCALE flag to be non-zero.