**Last Update:** 20 June 2018

RS5 Validation Guide

**Summary:** This document provides implementation steps to validate time accuracy with Windows 10 and Server 2016 scenarios.

Please Note: References to Windows Server 2016 can be used interchangeably with 2019

Table of Contents

[Overview 3](#_Toc510687718)

[Description 3](#_Toc510687719)

[Required Hardware 4](#_Toc510687720)

[Required Software 4](#_Toc510687721)

[Troubleshooting and Feedback 5](#_Toc510687722)

[Test Activities 5](#_Toc510687723)

[Activities 5](#_Toc510687724)

[Activity 1: Time Synchronization 5](#_Toc510687725)

[Activity 2: System Traceability 7](#_Toc510687730)

[Activity 3: SCOM Monitoring 8](#_Toc510687732)

[Appendix 9](#_Toc510687734)

[System Traceability Event Log 9](#_Toc510687735)

[SCOM Management Pack 11](#_Toc510687736)

[Importing Management Pack 11](#_Toc510687737)

[Override Time Accuracy Out of Range rule settings 12](#_Toc510687738)

[Observing an Exception 13](#_Toc510687739)

[Enabling Software Timestamping 15](#_Toc510687740)

[Measuring a System Under Test 15](#_Toc510687741)

[Measurement Prerequisites 15](#_Toc510687742)

[Data Gathering and Analysis 17](#_Toc510687743)

[Manual Data Analysis 19](#_Toc510687744)

[Troubleshooting HighAccuracyConfig.ps1 20](#_Toc510687745)

[Troubleshooting HighAccuracyCollection.ps1 20](#_Toc510687746)

# Overview

Many sectors require accurate time. The financial sector requires precise and accurate time so that transactions are properly timestamp to 50ms, 1ms or even 100µs accuracy. Windows 10 and Windows Server 2016 can maintain a precise and accurate system clock benefitting both physical and virtual workloads, including non-Windows OS guests such as Linux and OpenBSD. The Windows Server 2019 (RS5) and the corresponding Windows 10 release continues to improve windows precision and makes system time traceability easy by leveraging System event logging.

# Description

Windows Server 2016 can deliver accurate and precise time, provided you have the proper infrastructure. High accuracy requires a precise reference time source, like GPS backed hardware, to provide a stable signal as a source for the domain, or targeted systems.

The benefits of a Windows solution with a highly accurate time source are:

* Both physical and virtual machines can maintain accurate time < 1ms.
  + Includes Windows Server 2016 and Windows 10 (RS1/1607 or later)
  + Includes Linux and OpenBSD virtual machines that contain the Microsoft updates allowing the Linux machine to synchronize time from a Hyper-V host (RS1/1607 or later)
  + Includes Windows Server Containers
* Domain members automatically receive accurate time, discovering the best time architecture automatically
  + No third-party software is needed, lowering the cost of ownership and minimizing the number of products you need to administer and maintain
  + Security and maintenance is simplified as you no longer track 3rd party software updates
* RS5 contains event logging that runs continuously for the Windows Time service to assist with government traceability requirements that require clock sources are traceable to UTC and that the clock is continually being synchronized
  + Always on logging helps faster triage of identified issues and reduces configuration requirements
  + Logs can be archived for later analysis using your existing SIEM solution
* SCOM lets you monitor accuracy on critical systems
  + SCOM can be configured to alert when a machine does not meet expected thresholds
  + Quickly track down systems with issues

# Required Hardware

This scenario requires the following hardware and features for successful completion:

|  |  |
| --- | --- |
| Hardware | Description |
| Windows 10 or Windows Server 2016 (RS5) Domain or Standalone (includes virtual machines and their host systems). | Note: To test virtual systems, hosts must be able to run Hyper-V. |
| GPS hardware or Time appliance | A stable signal requires a stable source. 1ms or greater accuracy is not possible if the upstream clock source isn’t at least as accurate.  Note: We are still interested in your observations if you do not have a highly accurate time source. Also, please contact us if you would like examples of appropriate time sources. |

# Required Software

|  |  |
| --- | --- |
| Software | Description |
| Windows 10 or Windows Server 2016 (RS5) | Note: The time accuracy improvements were included in Windows 10 version 1607 (Redstone 1) and Windows Server 2016 version 1607 (Redstone 1).  Each subsequent version includes additional improvements to our precision and accuracy. |
| System Center Operations Manager | Required only to test the SCOM alerting |
| Measurement Tools | You’re welcome to leverage your own tools and we have some tools [located on GitHub](https://github.com/Microsoft/Windows-Time-Calibration-Tools) |

# Troubleshooting and Feedback

If you encounter any challenges during your validation, please use one of the contact methods identified on the title page of this document.

Also, please submit your feedback!

# Test Activities

Below are the activities that we would like validation:

1. **Time Synchronization** – 1ms accuracy running your applications on Windows 10 or Windows Server 2016 physical or virtual systems.
2. **System Traceability** – Meet regulatory obligations for logging and archiving Windows Time service event logs.
3. **Alerting with SCOM** – Get alerted via SCOM when a defined accuracy is exceeded.

# Activities

## Activity 1: Time Synchronization

This activity will test accurate time synchronization to 1ms accuracy. Please complete one or more of the following tasks and provide feedback through one of the channels on the title page of this document.

* Task 1 – 1ms with Windows Physical machine (desktop or server)
* Task 2 – 1ms with Windows VM (desktop or server)

### Activity 1: Minimum Requirements

This section outlines the information needed to setup this activity for a Windows Physical or Virtual Machine

* Download the latest Insiders or TAP build available
  + These activities require version Redstone 1 (version 1607) of both Windows 10 or Server 2016 or later. Please consider using the latest available build as some of the features (for example, Software Timestamping) are available only in RS4 1803 and higher (Server 2019)
  + Redstone 5 server builds will be the latest Long-Term Servicing Branch (LTSC) and is now known as Windows Server 2019 – For purposes of this document, this terminology will be used interchangeably.
  + Hyper-V Hosts that run virtual workloads for testing must also be configured using the high accuracy configuration noted later in this document
* Each System Under Test (SUT)
  + Should be no more than stratum 3 from your time source. The command **w32tm /query /status** can be used to obtain the stratum of a system.



* + Should be configured with the High Accuracy configuration located under:

**HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\W32Time\Config**

|  |  |
| --- | --- |
| Registry Entry | Value |
| MinPollInterval | 6 |
| MaxPollInterval | 6 |
| UpdateInterval | 100 |
| FrequencyCorrectRate | 2 |

* + Configure your SUT to point to a highly accurate time source using the following command:

**w32tm /config /manualpeerlist:master\_clock1,0x8 /syncfromflags:manual /reliable:yes /update**

Alternatively, you may choose to utilize the **HighAccuracyConfig** scripts provided with this guide. More information is provided under [Measuring a System Under Test](#_Measuring_a_System)

### Activity 1: 1ms Accuracy with a Windows Physical machine

With the required changes noted in the [minimum requirements](#_Activity_1:_Minimum), a physical Windows system running Windows Server 2019 should realize high accuracy without additional changes.

Note: The SUT will be gradually brought into accuracy. This may take several hours depending on the SUT.

### Activity 1: 1ms accuracy with Windows VM (desktop or server)

To synchronize time on virtual systems, the following options are available:

* Directly configure your virtual system to a highly accurate time source
* Utilize the Virtual Machine Integration Components (VMIC) which are enabled by default

For simplicity we recommend that you utilize only one of the above options at a time and will show you how to disable the other.

#### Utilize the VMIC for Time Synchronization

There is no configuration required to enable the inbox virtual machine integration components. The OOB configuration will natively provide time synchronization through the VMIC. However, in order to provide high accuracy time from your host to a virtual machine, the host and virtual machine must both be configured with high accuracy settings noted earlier.

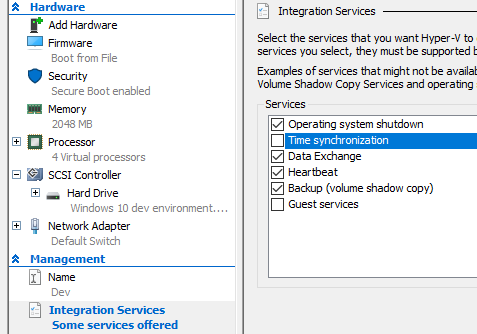
Note: Your Hyper-V host is considered a stratum and Windows 10 (Client Hyper-V) and Server 2016/Server 2019 will count the host in the displayed stratum when using the command **w32tm /query /status**

#### Configure your system directly to a Time Source

In this scenario where you are configuring your system directly to a time source, you may choose to disable the VMIC to avoid confusion and configuration challenges.

To do this perform the following steps:

* In the VM add the following registry entry
  + HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\W32Time\TimeProviders\VMICTimeProvider
  + Name: Enabled
  + Type: Reg\_DWORD
  + Value: 0
* Shut the VM Down
* In the VM Settings uncheck the “Time synchronization” box and click OK

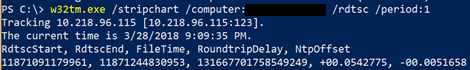


* Power on the virtual machine

### Simple Accuracy Observation

You can use the inbox tool w32tm.exe to observe basic accuracy from the SUT. Run the following command from your SUT and target your clock source.

**w32tm.exe /stripchart /computer:*ReferenceClock* /rdtsc /period:1**



The primary column we’re interested in is the NtpOffset which is shown in seconds.

* If the NtpOffset shows a negative number, the reference clock is behind the local clock
* If the NtpOffset shows a positive number, the reference clock is ahead of the local clock

In the example above, the local system (SUT) is ahead of the reference clock by 5.1 milliseconds.

Note: RoundtripDelay is also a significant number. The larger the RoundtripDelay, the harder it becomes to calculate NtpOffset. In the example above, the RoundtripDelay is 54 milliseconds.

For a longer running test please see [Measuring a System Under Test](#_Measuring_a_System)

### Activity 1: Accuracy with Software Timestamping

In 1803 (RS4) we introduced a new capability known as software timestamping. That calculates the additional delay added to time samples caused by the network stack (excluding the miniport).

To test software timestamping on your system accuracy, enable one machine with software timestamping and one machine without. Both systems should be setup with the additional high accuracy configuration noted earlier in this document. Then simultaneously run the following command from each machine. For a more accurate test, see [Measuring a System Under Test](#_Measuring_a_System) and perform the steps against both systems simultaneously.

**w32tm.exe /stripchart /computer:*ReferenceClock* /rdtsc /period:1**

To enable Software Timestamping, please see [Enabling Software Timestamping](#_Enabling_Software_Timestamping)

## Activity 2: System Traceability

This activity will validate that the Windows Time service logging meets your needs for system traceability. The complete list of logs is available under [System Traceability Event Log](#_System_Traceability_Event).

System time event logging is enabled by default and is in the Event Viewer under the Operational channel at: **Applications and Services Logs** > **Microsoft** > **Windows** > **Time-Service**

Please complete the following tasks:

* Review the available Time-Service messages in the SUT’s event logs
* Validate that the logs can be consumed using your existing log archival solution (SIEM)
* View the log entries that are recorded, as well as the list (in section 7.1) to validate it contains the information needed to meet regulations around system traceability.

### Activity 2: Questions

* Does the logging contain all the information you need for system traceability?
* Did the event log integrate with your SIEM/log archiving solution as expected?
* What improvements could be made to this logging?

## Activity 3: SCOM Monitoring

In this activity, use System Center Operations Manager to monitor a machines accuracy and alert and record when it is out of bounds, based on your requirements. The task assumes you have SCOM setup already.

For information on importing this management pack, please see [Importing a SCOM Management Pack](#_Importing_Management_Pack)

Modify the following settings to meet your requirements:

* Interval Seconds – the number of time between samples in seconds.
* Number of Samples – The number of samples that are above your threshold before an alert occurs.
* Threshold in Milliseconds

For information on how to toggle these settings, please see [Override Time Accuracy Out of Range rule settings](#_Override_Time_Accuracy)

For information on how to observe an exception, please see [How to observe an exception](#_How_to_observe)

### Activity 3 – Questions

* Were you able to use the MP to monitor your windows environment using the SCOM?
* Are there any addition rules that you would like to add?

# Appendix

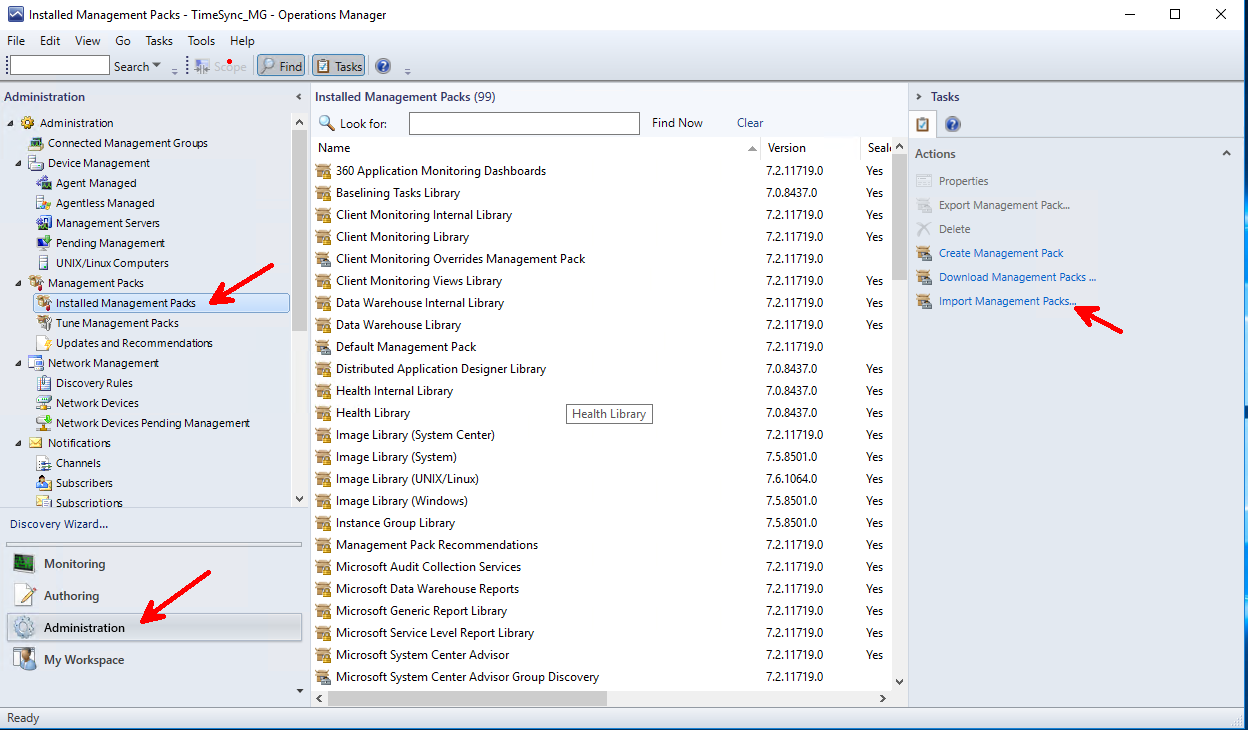
## System Traceability Event Log

|  |  |  |  |
| --- | --- | --- | --- |
| Event # | Event Description | Details | Throttling Mechanism |
| 257 | Service Start | Occurs at W32time Startup | None |
| 258 | Service Stop | Occurs at W32time Shutdown | None |
| 259 | List of time sources(s) used by NTP Client | NTP Client periodically logs its current list of time sources and its most accurate time source. | Logged once every 8 hours |
| 260 | Time service configuration and status | W32time periodically logs its configuration and status. This is the equivalent of calling:  **w32tm /query /configuration /verbose**  OR  **w32tm /query /status /verbose** | Logged once every 8 hours |
| 261 | System Time is set | This logs each instance when System Time is modified using SetSystemTime API. | None.  This should happen rarely on systems with reasonable time synchronization, and we want to log it each time it occurs.  We ignore *TimeJumpAuditOffset* setting while logging this event since that setting was meant to throttle events in the Windows System event log. |
| 262 | System clock frequency adjusted | System clock frequency is constantly modified by W32time when the clock is in close synchronization. We want to capture “reasonably significant” adjustments made to the clock frequency without overrunning the event log. | All clock adjustments below ***TimeAdjustmentAuditThreshold***(min = 128 part per million, default =800 part per million) are not logged.  2 PPM change in clock frequency with current granularity yields 120 µsec/sec change in clock accuracy.  On a synchronized system, majority of the adjustments are below this level. If finer tracking is desired, this setting can be adjusted down and/or PerfCounters should be used. |
| 263 | Change in the Time service settings or list of loaded time providers | Re-reading W32time settings can cause certain critical settings to be modified in-memory, which can affect the overall accuracy of the time synchronization.  W32time logs each occurrence when rereading its settings which gives the potential impact on time synchronization. | None.  This event occurs only when an admin or GP update changes the time providers and then triggers W32time. We want to record each instance of change of settings. |
| 264 | Change in time source(s) used by NTP Client | NTP Client records an event with the current state of the time servers/peers when a time server/peer changes state (Pending ->Sync Sync -> unreachable or other transitions) | Max frequency – only once every 5 minutes to protect the log from transient issues and bad provider implementation. |
| 265 | Time service source or stratum number changes | W32time Time Source and Stratum Number are important factors in time traceability and any changes to these must be logged. If W32time has no source of time and it is not configured as a reliable time source, then it will stop advertising as a time server and by-design respond to requests with some invalid parameters. This event is critical to track the state changes in an NTP topology. | None |
| 266 | Time re-synchronization is requested | This operation is triggered:   * + - When network changes occur     - System returns from connected standby/hibernation     - When we didn’t sync for a long time     - Admin issues the resync command   This operation results in immediate loss of fine-grained time sync accuracy because it causes NTP client to clear its filters. | Max frequency - once every 5 minutes.  It is possible that a bad network card (or a poor script) can trigger this operation repeatedly and result in logs getting overwhelmed. Hence the need to throttle this event.  Note that accurate time sync takes far more than 5 minutes to achieve, and throttling does not lose information about the original event that resulted in loss of time accuracy. |

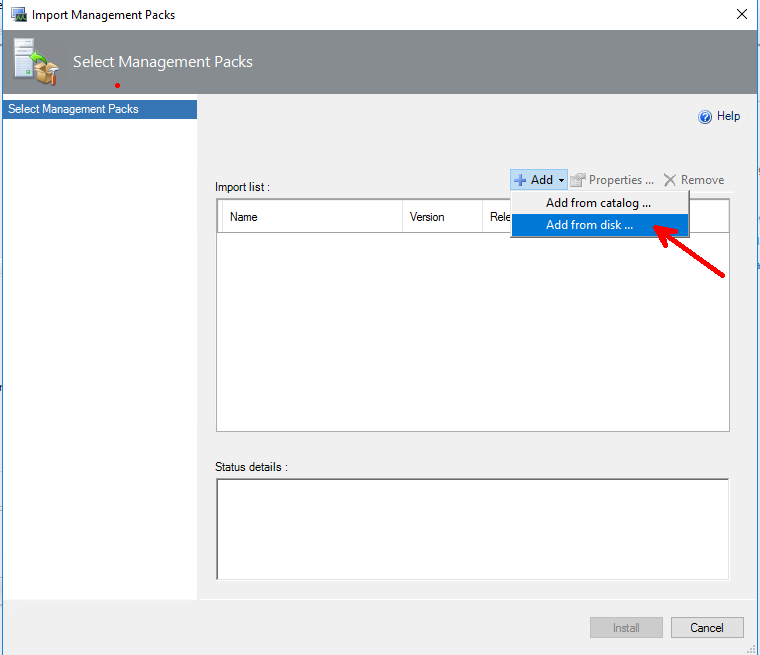
## SCOM Management Pack

### Importing Management Pack

Select the administrative mode in Operations Manager. Select the Install Management Packs node and choose Import Management Packs on the right.

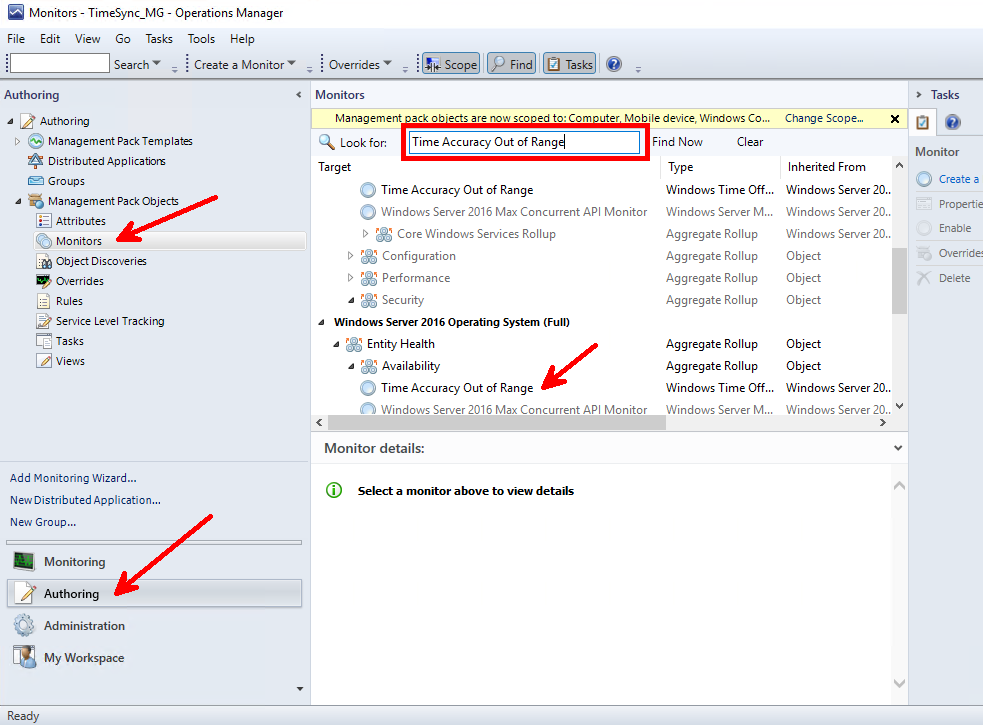


Choose Add from disk and install the Microsoft.Windows.Server.2016.Monitoring.mp along with its dependencies.

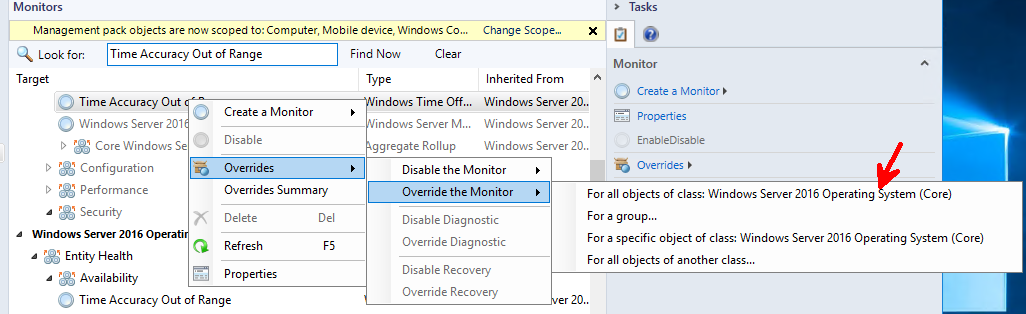


### Override Time Accuracy Out of Range rule settings

Use the Authoring mode and select Monitors. Search for “Time Accuracy Out of Range”.

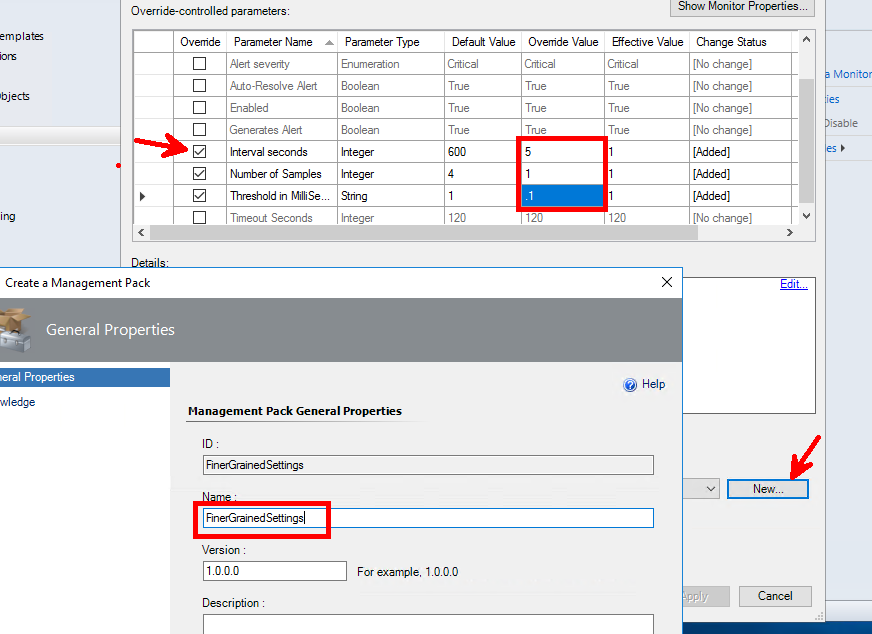


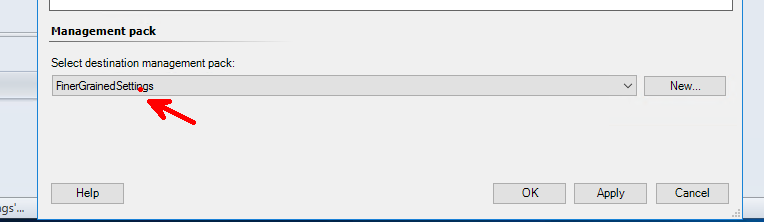
Right click the rule and select Overrides and follow the menu shown below to bring up the editor.



Then modify the settings as you wish. The defaults specify that it will wait for 4 sampled events that exceed 1 ms, every 600 seconds, and then raise an alert.

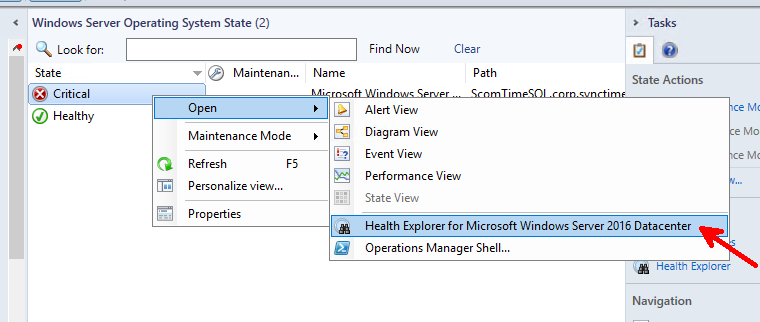
In the example below, I changed the settings by checking the Override check boxes so that now we raise an event when just one sample exceeds 100us that it checks every 5 seconds. Once you make a change you need to save it to a destination MP using a name you created, in my case FinerGrainedSettings.



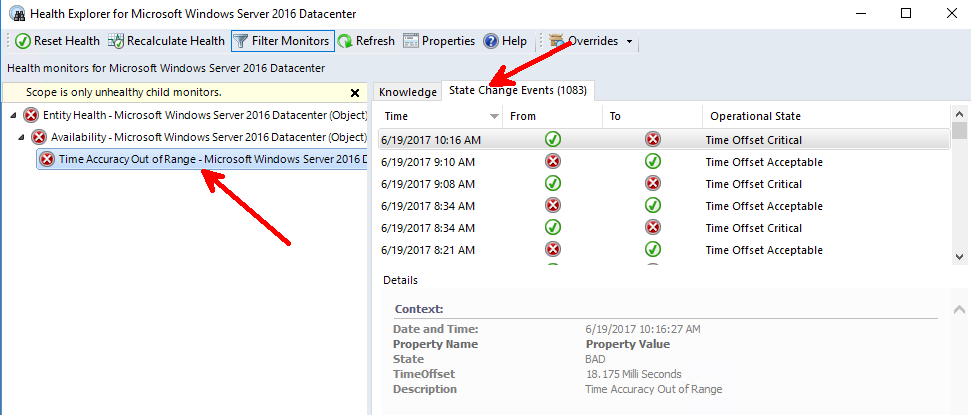


### Observing an Exception

Once a monitor machines is in a problem state, you should see it indicated on the Monitoring mode under the Microsoft Windows Server->Windows Server Operations System State. You can view more specifics by selecting a system and right clicking Open->Health Explorer for Microsoft Windows Server 2016 Datacenter.



You can see more detail by viewing the selecting the Time Accuracy Out of Range rule, and viewing the State Change Events, which show a record of the systems events.



## Enabling Software Timestamping

One additional feature to further increase your time accuracy, is by enabling software timestamping. This feature calculates the delay added to a time sample by the networking stack (excluding the miniport). You enable this setting for each NIC processing the packets.

**Note:** This feature is not enabled by the **HighAccuracyConfig** scripts provided with this guide.

**To enable software timestamping:** Please refer to the [Software Timestamping Validation Guide](https://github.com/Microsoft/SDN/blob/master/FeatureGuide/Validation%20Guide%20-%20RS5%20-%20Software%20Timestamping.docx)

## Measuring a System Under Test

In general, to validate the accuracy and stability, frequent samples are taken over long periods to determine the accuracy. The longer tests are run, the better the data will be. It’s recommended to run for a minimum of 24 hours, and preferably longer.

A simple test requires three systems. The System Under Test (SUT), the high accuracy reference clock (REF), and the observation machine (OBS). From the OBS you will measure the SUT and the REF simultaneously. The data from the SUT is then compared against the data from the REF. Since the network will affect the measurements, you want to minimize the number of hops when collecting the data. Ideally both the SUT and REF clock are located near each other on your network.

**Note**: Following configuration of the SUT for high accuracy the Windows Time service takes time to discipline the clock. We recommend waiting 2 – 4 hours prior to measurement.

### Measurement Prerequisites

For measurement purposes, we require that the system respond to NTP requests. As such, we need to enable the NTP Server on each system being measured. To do this:

1. High Accuracy Configuration noted earlier in this document
2. Enable the NTP Server Firewall rule using PowerShell

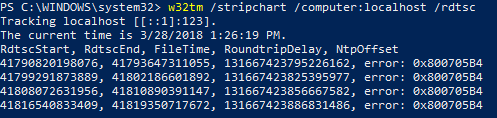
**New-NetFirewallRule -DisplayName "Allow in NTP" -LocalPort 123 -Protocol UDP -Action Allow**

1. Enable the NTP Server so the system can respond to NTP messages using the following three PowerShell commands:

**Set-ItemProperty -Path hklm:\SYSTEM\CurrentControlSet\Services\w32time\TimeProviders\NtpServer -Name Enabled -value 1**

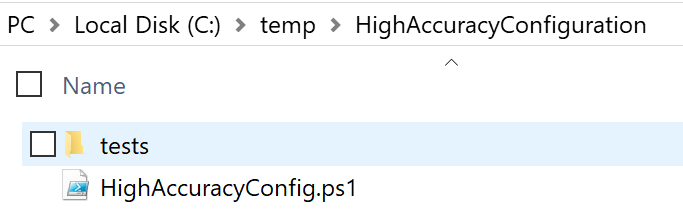
**Restart-Service W32Time**

NtpOffset **error: 0x800705B4** is an indication that NTP server was not available. If you are seeing this error consistently, it is likely that the NTP Server is not enabled (**Set-ItemProperty** above requires service restart), the firewall is not allowing the traffic (review Windows Firewall logs), or some physical network related issue is preventing completion of the query (verify consistent ping connectivity).

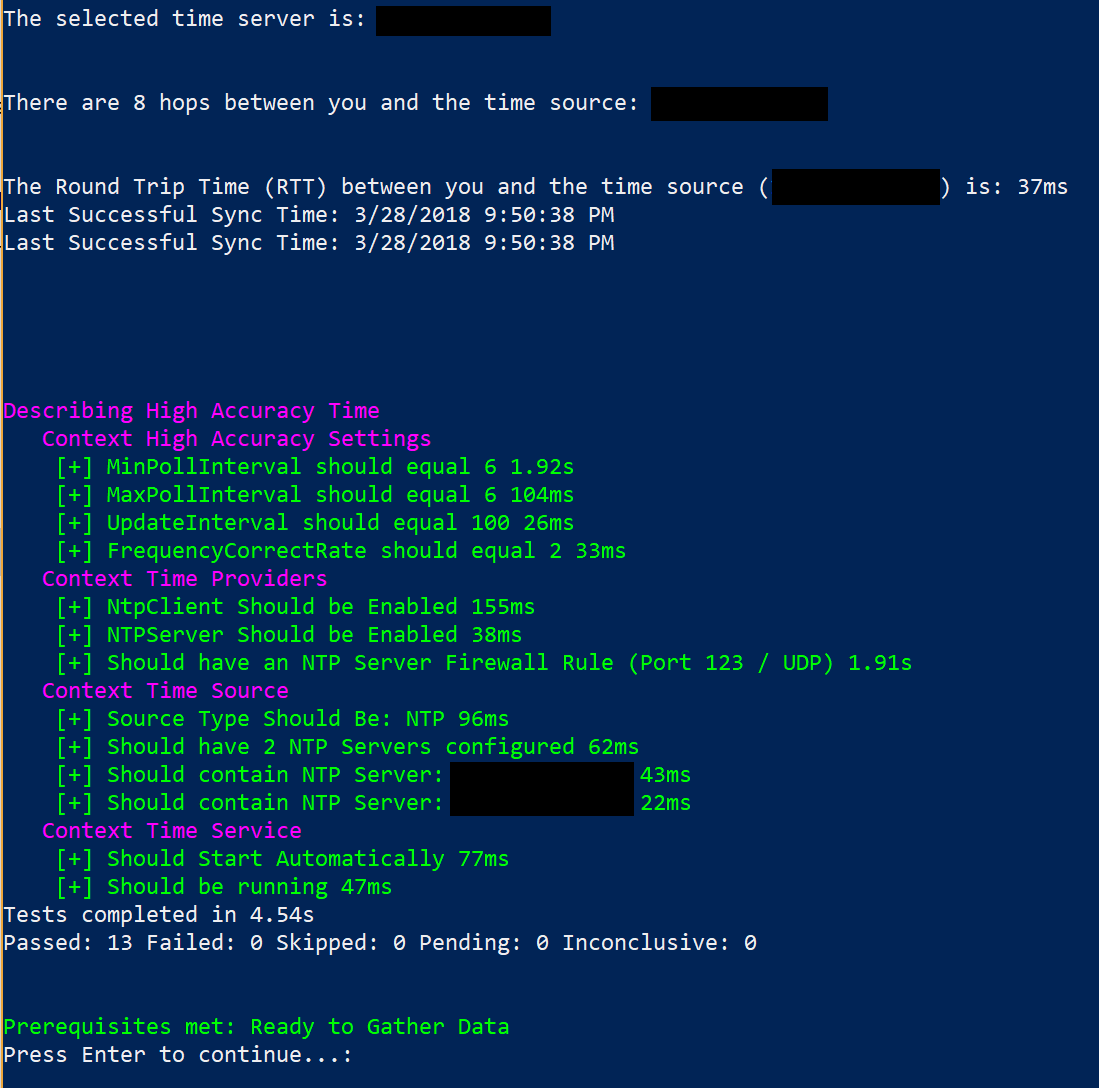


Alternatively, you may choose to use the **HighAccuracyConfig.ps1** script provided with this. If you choose to use the HighAccuracyConfig, please verify that the tests report that ***Prerequisites met: Ready to gather data****.* If there is a failure, the tests should indicate a possible reason for the failure. Please use the contact information on the title page if you have any questions.

To run the HighAccuracyConfig.ps1, extract the folder contents a single folder for example



Next, right-click and “Run with PowerShell.” Below is output from a system configured with two NTP servers



**Note**: The script **HighAccuracyConfig** script is provided as an example and will require modification for your test. For example, you must update the NTPServers section of the script. Please use the contact information on the title page if you have any questions.

### Data Gathering and Analysis

From an admin system, run the following command (example below) for the SUT. In addition, you must run this command against the REF system at the same time.

For example, with a system under test named **SUT1** and a reference clock of **Ref1** run the following commands from your admin station simultaneously. Note: You should run these commands at the same time from separate command prompts.

**w32tm.exe /stripchart /computer:SUT1 /rdtsc /period:1 /samples:900 > c:\temp\SUT1.out**

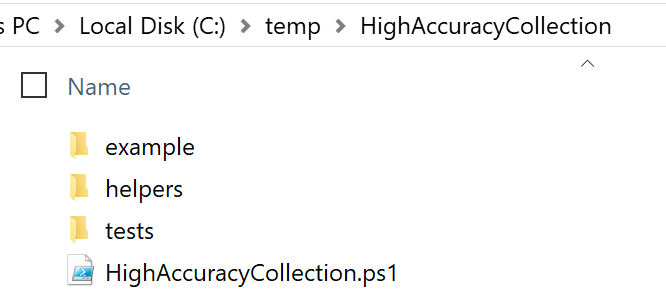
**w32tm.exe /stripchart /computer:Ref1 /rdtsc /period:1 /samples:900 > c:\temp\Ref1.out**

The command above captures 1 sample every second (period) for 15 minutes (samples in seconds) and outputs the data to a text file. For a more complete test, capture the data over approximately 24 hours.

To help with this and enable scale, we have a sample monitoring service available in our [GitHub tools](https://github.com/Microsoft/Windows-Time-Calibration-Tools/tree/master/NtpMonitoringService) called the [NTP Monitoring Service](https://github.com/Microsoft/Windows-Time-Calibration-Tools/tree/master/NtpMonitoringService) that allows larger scale monitoring. While unnecessary for this smaller test, you may choose to expand your test; instructions are available on the site.

Alternatively, you may choose to use the **HighAccuracyCollection.ps1** script provided with this guide. If you choose to use the HighAccuracyCollection, please verify that the tests report that ***Prerequisites met: Ready to gather data****.* If there is a failure, the tests should indicate a possible reason for the failure. Please use the contact information on the title page if you have any questions.

To run the HighAccuracyConfig.ps1, extract the folder contents to a single folder on the OBS system. This is a tertiary system used to perform measurement of the SUT and REF systems.

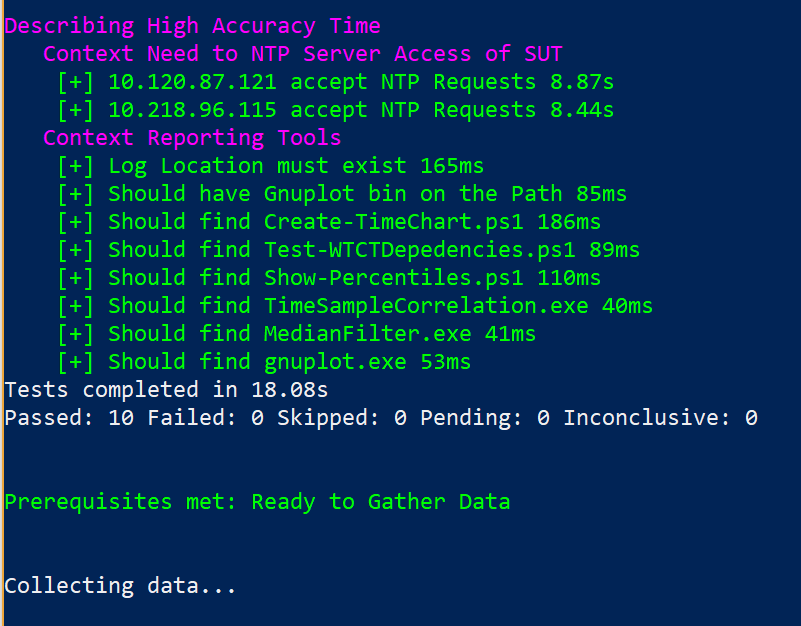


Next install [GnuPlot](http://gnuplot.info/download.html) on the OBS system. This will be used to generate a graphical picture from the data.

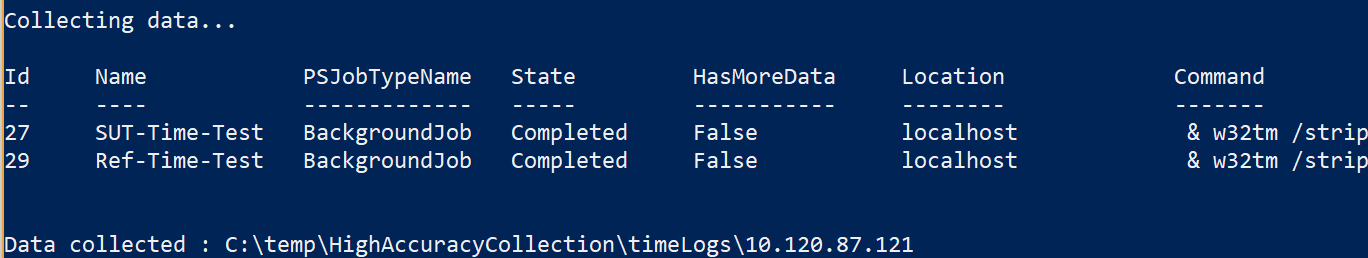
Run the **HighAccuracyCollection.ps1** script from an elevated PowerShell prompt and specify the **SUT** parameter and the **NTPServer** parameter



The Script will test the OBS system for the necessary prerequisites and will report if the prerequisites are met. Please resolve any error found prior to continuing.



Data is collected using background PowerShell jobs. When the configured samples have been completed, the data will be collected in a folder named timelogs\SUT



In the HighAccuracCollection\timeLogs\SUT folder you will find percentiles.txt and Graphdata\SUT.png

**Percentiles.txt**

This file shows the percentage of samples that were below a specific accuracy for example:

* + - **68th Percentile = 458.3us** – 68% of the samples taken from the SUT were under 458.3us
    - **95th Percentile = 796.8us** – 95% of the samples taken from the SUT were under 796.8us
    - **99.7th Percentile = 1242.6us** – 99.7% of the samples taken from the SUT were under 1242.6us

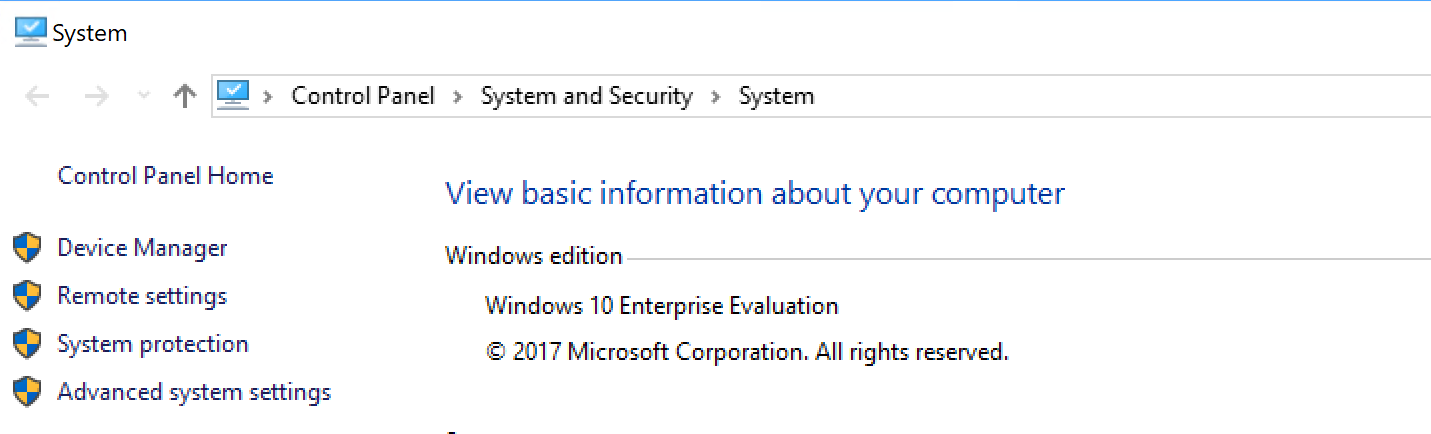
**Graphdata\SUT.png** – This presents a graphical readout of the UTC delta and Round Trip Time

**Note**: The **HighAccuracyCollection** script is provided as an example and will require modification for your test. For example, the prerequisite validation tests provided may not be pertinent to your environment. Please use the contact information on the title page if you have any questions.

### Manual Data Analysis

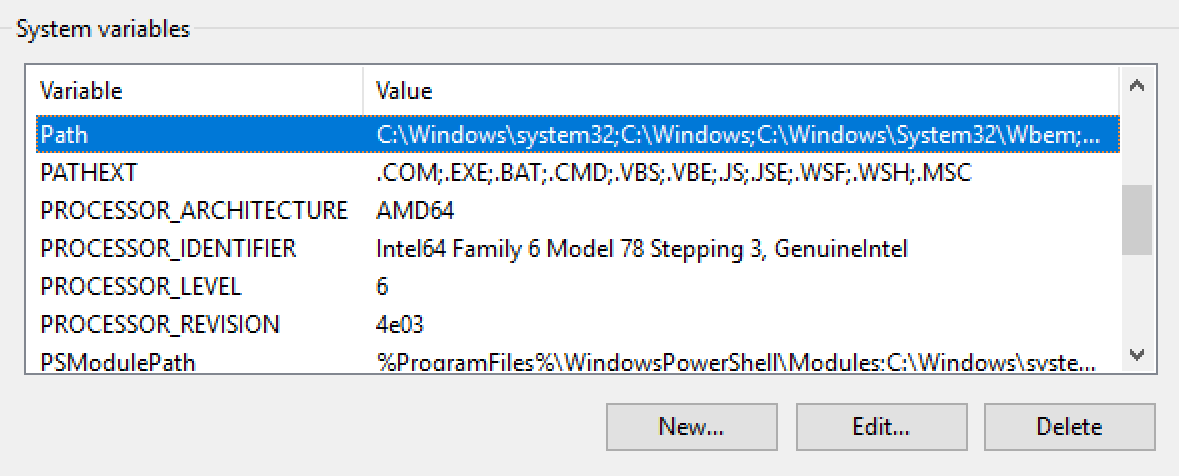
If you manually collected data (without the use of the HighAccuracyCollection script) you can still analyze the data. On the OBS machine (where the result text files exist) install [GnuPlot](http://gnuplot.info/download.html).

You must also put gnuplot.exe on the system path

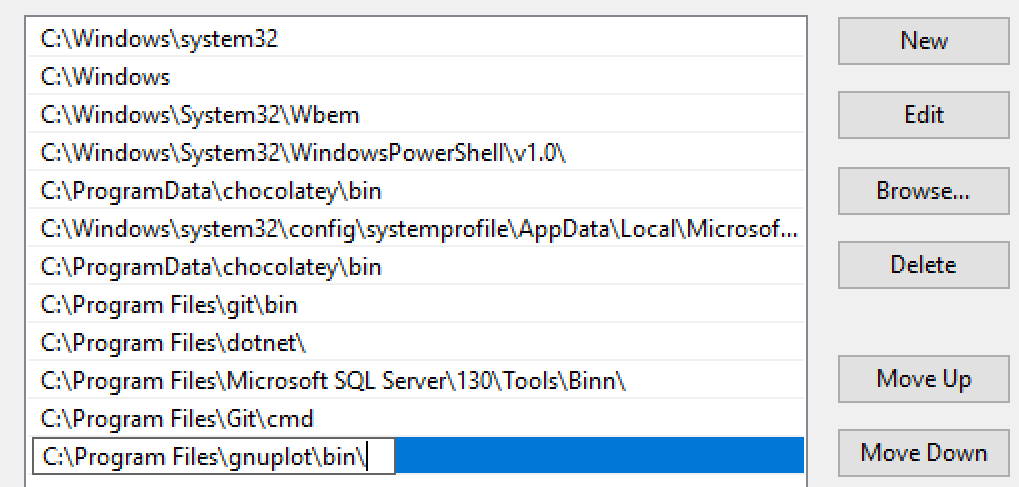


Click **Advanced System Settings**, then **Environment Variables**

Under **System variables**, select **Path**, then click **Edit…**



Click **New**, then click add the bin path for gnuplot



Repeat the steps to add the path to the HighAccuracyCollection\helpers folder included with this guide. These scripts contain the required components to analyze the data.

Lastly, run **Create-TimeChart.ps1 -Name SUT -ReferenceClock SUT**

**Create-TimeChart.ps1** is part of the [Windows Time Calibration Tools](https://github.com/Microsoft/Windows-Time-Calibration-Tools/tree/master/Scripts) Github project.

**Note**: Please use the contact information on the title page if you have any questions.

### Troubleshooting HighAccuracyConfig.ps1

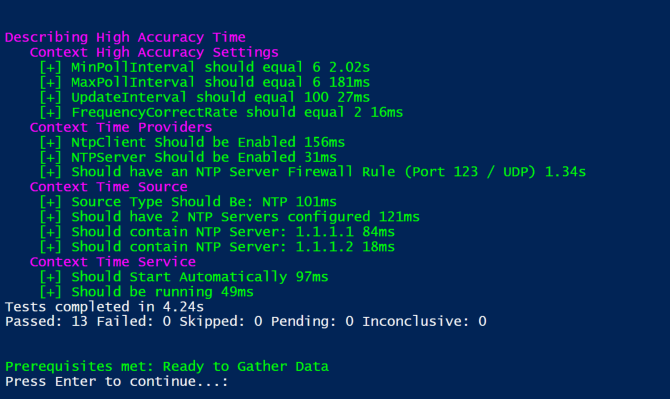
**Script Elevation**

Both configuration scripts require elevation. The script will attempt to do this automatically for you but may require interaction or appropriate login based on your environment.

**Failing Tests**

Tests are provided to simplify troubleshooting of requirements. However, based on your environment, these may not be necessary or appropriate. Tests are generated using the inbox Pester PowerShell Module for your convenience.

Example passing tests:

Verify that **Prerequisites are met: Ready to Gather Data** message is seen below

Failing Tests: Tests are provided as an example to indicate a failure that may prevent proper collection or analysis. Not every test will apply to every scenario. If you have questions in this area, please contact us.

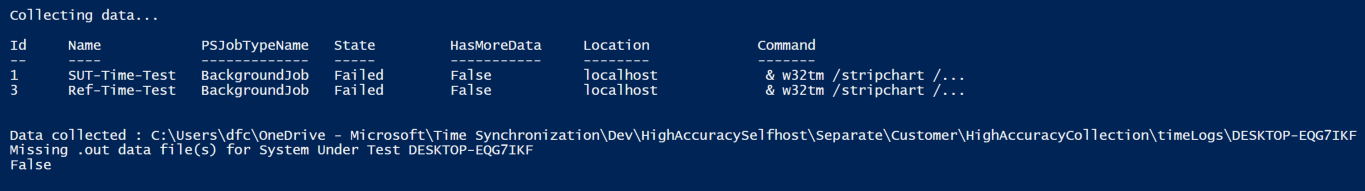
### Troubleshooting HighAccuracyCollection.ps1

**Script Elevation**

Both configuration scripts require elevation. The script will attempt to do this automatically for you but may require interaction or appropriate login based on your environment.

**Missing .Out data file(s) due to IPv6**

This error is most commonly caused by an IPv6 system in the mix. While high accuracy configuration is supported on IPv6 systems, the files cannot be automatically created using the scripts provided. Either enter the IPv4 address of the system, or manually create the w32tm /stripchart data as shown under [Data Gathering](#_Data_Gathering).



**Missing .Out data file(s) due to path issues**

This error can also be observed due paths where the files could not be created. For a simple fix, move the HighAccuracyCollection folder to c:\temp\HighAccuracyCollection and try again.