**Last Update:** 1 July 2018

RS5 Validation Guide

**Summary:** This document provides implementation steps for an IT Pro to validate leap seconds with Windows 10 and Server 2016 scenarios.

Table of Contents

[Overview 3](#_Toc518314356)

[Description 3](#_Toc518314357)

[Required Software 4](#_Toc518314358)

[Test Activities 4](#_Toc518314359)

[Before You Begin 4](#_Toc518314360)

[Activities 6](#_Toc518314361)

[Activity 1: Modify taskbar clock to display seconds 6](#_Toc518314362)

[Activity 2: Modify regional settings to display seconds 6](#_Toc518314363)

[Activity 3: Insert Positive Leap Seconds 7](#_Toc518314364)

[Activity 4: Observe the leap second 9](#_Toc518314365)

[Activity 5: Insert Negative Leap Seconds 11](#_Toc518314366)

[Activity 6: Test an Application 11](#_Toc518314367)

[Activity 7: Test a System 12](#_Toc518314368)

# Overview

Many sectors require accurate time. The financial sector requires accurate time so that transactions are properly timestamp to 50ms, 1ms or even 100µs accuracy. In addition, these regulations require that time be compliant with UTC. Windows 10 and Windows Server 2019 is Leap Second aware. This enables Windows to stay within their time-accuracy boundaries through a leap second and apply leap seconds in a compliant manner.

# Description

A leap-second is a 1-second adjustment to UTC.  As the earth’s rotation slows, [UTC](https://en.wikipedia.org/wiki/Coordinated_Universal_Time) (or atomic time) diverges from [mean solar time](https://en.wikipedia.org/wiki/Solar_time#Mean_solar_time) or astronomical time.  Once UTC has diverged by at most .9 seconds, a [Leap Second](https://en.wikipedia.org/wiki/Leap_second) is planned and inserted to keep UTC in-sync with mean solar time.  There have been 27 leap seconds so far inserted into UTC.

Usually leap seconds are added when UTC is ahead of UT1 by 0.4 seconds or more. There have been leap seconds about every year and a half (on average) ([NIST FAQ](https://www.nist.gov/pml/time-and-frequency-division/leap-seconds-faqs)).

International Earth Rotation and Reference Systems Service (IERS) oversees the announcement of Leap Seconds. They release several [bulletins](https://www.iers.org/IERS/EN/Publications/Bulletins/bulletins.html) used in this process.

Bulletin C ([IERS leap second announcements](https://www.iers.org/IERS/EN/Publications/Bulletins/bulletins.html)) is mailed every six months, either to announce a time step in UTC, or to confirm that there will be no time step at the next possible date. The most recent Bulletin C was released in January 2018 and announced no Leap Second would be inserted; the next Bulletin is expected sometime in early July.

Most importantly, FINRA, the regulatory arm of the Securities and Exchange Commission, requires that clocks “must be synchronized within a 50-millisecond tolerance of the NIST clock. This tolerance includes all of the following:

1. The difference between the NIST standard and a time provider's clock;
2. Transmission delay from the source; and
3. The amount of drift of the member's clock.

Considering these requirements (ESMA, the counterpart in the EU, has more stringent requirements), leap seconds must be properly applied, and accuracy maintained during this application. Windows Server 2019 will apply Leap Seconds properly and maintain the appropriate level of accuracy during a leap second.

# Required Software

|  |  |
| --- | --- |
| Software | Description |
| Windows 10 or Windows Server 2019 (RS5) | Leap Second improvements are only found in Redstone 5 versions of Windows 10 (v1809) or Windows Server 2019 |

# Test Activities

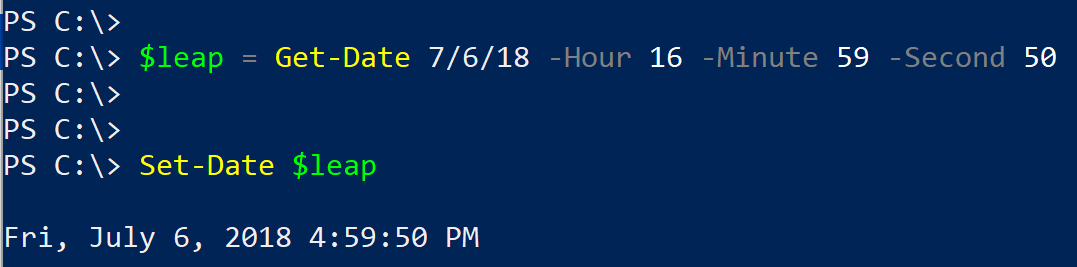
Below are the activities that we would like validation:

1. **Modify taskbar clock to display seconds** – Enable the taskbar clock to display seconds
2. **Modify regional settings to display seconds**– Enable file time creation and modification dates to display seconds
3. **Insert Positive Leap Seconds** – Add positive leap seconds
4. **Observe the leap second** – Verify the leap seconds applied properly
5. **Insert Negative Leap Seconds** – Add negative leap seconds
6. **Test an Application** – Verify functionality of an application or specific actions through a leap second
7. **Test a System** – Verify functionality of a holistic system through a leap second

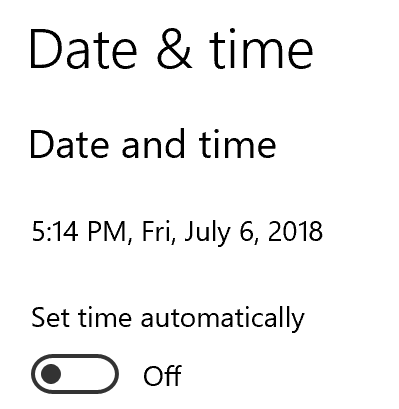
# Before You Begin

You should be aware of the following when testing leap seconds:

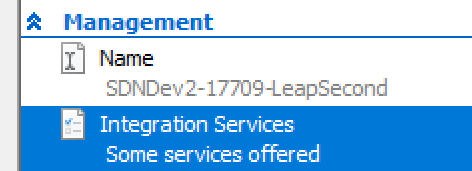
* Test leap seconds can be added using the w32tm command
* W32tm accepts input in UTC date and time format – e.g. Year-Month-Day**T**HH:MM:SS
* Leap Seconds can only be added at the end of an hour – E.g. **2018-7-6T16:00:00** is interpreted the same as **2018-7-6T16:59:59**
* Considering leap seconds can only be applied at the end of the hour, you may want to change the day, or time. Here are some programmatic options to change the date and time.
  + **Time** legacy command allows you to set the time in local time
  + **Date** legacy command allows you to set the date
  + **Set-Date** (PowerShell) allows you to change the date and time (in local time)

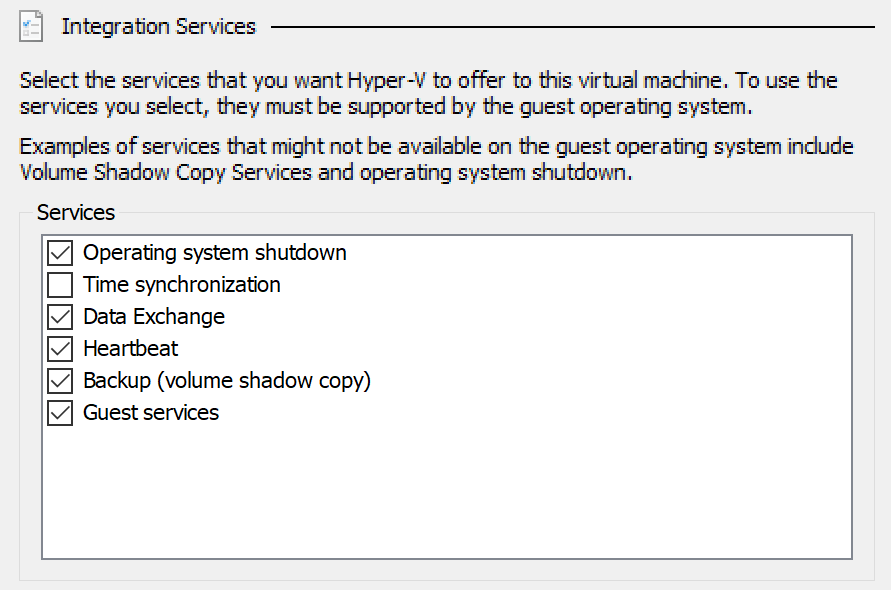


* If you’ve modified the day or time for this test, you may need to prevent the time service from resetting the time. To do this, disable the time service, or turn off the **Set time automatically** button in the **Date & Time** panel.



* If you’re testing in a virtual machine, you may also need to disable Virtual Machine Integration Component time provider. To do this,
  + Disable the **Time Synchronization** integration service in the virtual machine settings





* Some applications are known to calculate time incorrectly by assuming that there are always 60 seconds in a minute. Since leap seconds can change this behavior, they will improperly record the time during this event. For example (at the time of writing):
  + **.NET Framework** uses its own internal logic to determine what time it is and does not account for leap seconds. As a result, PowerShell, which relies on the .NET Framework, will not report the 61st second (number 60) when using Get-Date
  + **Event Viewer:** The date of the event will be incorrectly recorded. However, the event metadata will properly record the system time (showing the 60th second).

**Note:** These teams are working towards updating their software to use more appropriate math when handling leap seconds.

# Activities

## Activity 1: Modify taskbar clock to display seconds

In this activity, we will configure the taskbar clock to display seconds so that ‘test’ leap seconds can be observed. This is a visual test of leap seconds – Under normal scenarios, this activity is not required.

First, you must user settings that control the taskbar clock. To do this:

1. Right-Click on the start button and select **Run**
2. In the dialog box, enter **regedit** and click **OK**
3. If prompted for UAC elevation, click **OK**
4. In the registry editor, navigate to HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\Advanced
5. Create or modify the following registry entry:
   1. Type: Reg\_DWORD
   2. Name:ShowSecondsInSystemClock
   3. Value:1
6. Clock the registry and restart the computer
7. After restarting the computer, login with the same credentials (the registry change earlier was specific to the user). You should now see that the taskbar clock displays the seconds.

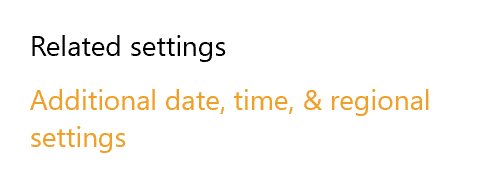


## Activity 2: Modify regional settings to display seconds

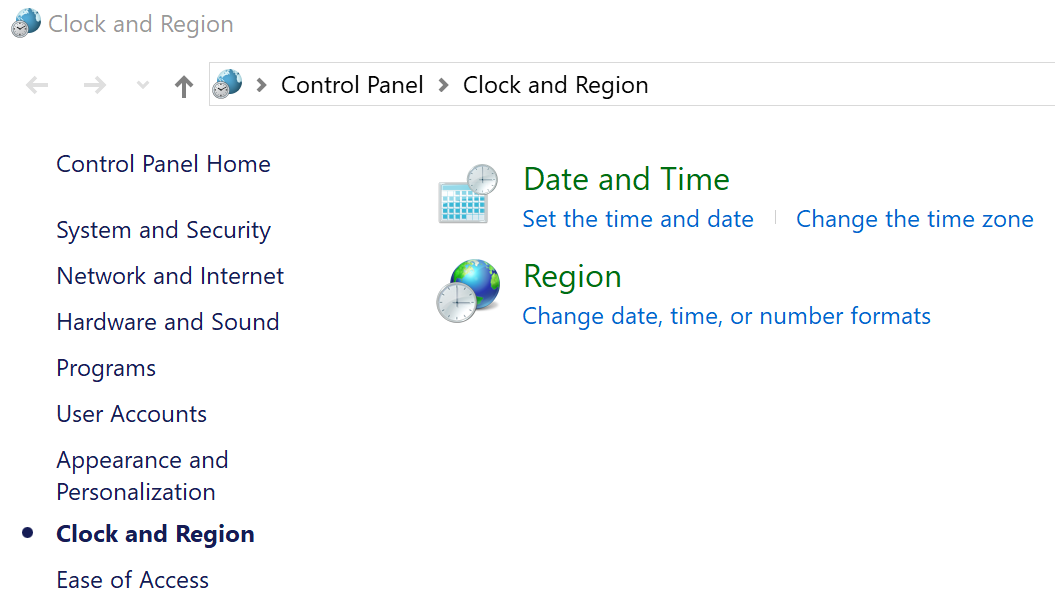
In this activity, we will configure file creation and modification times to display the seconds. This will allow us to verify that a leap second will be properly recorded by the file system if a log file is created or modified during this time.

You must modify the region settings to change from short date notation to long date notation. To do this:

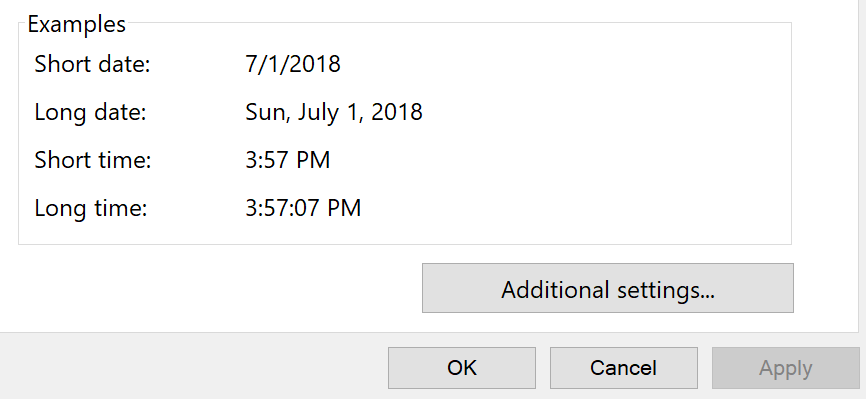
1. Right-Click on the start button and select **Settings**
2. In the settings panel, select **Time & Language**
3. In the right-hand side, under *Related settings*, select **Additional date, time, & regional settings**



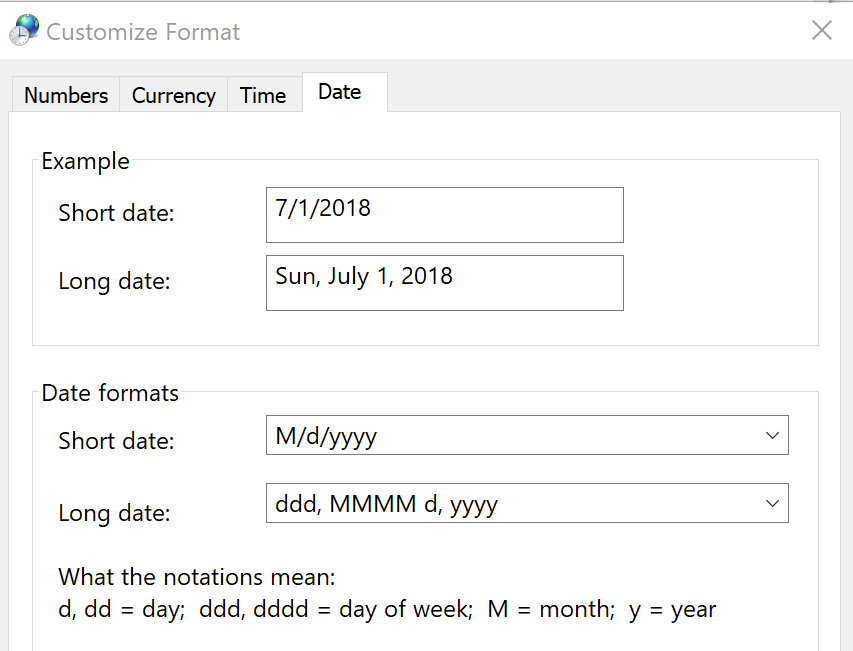
1. Under *Region* select **Change date, time, or number formats**



1. Select **Additional Settings**



1. On the *Date* tab modify the *Long date* to include only 3 lowercase d’s so the line appears as shown in the picture below



1. Apply and Exit the control panels

## Activity 3: Insert Positive Leap Seconds

In this activity, we will insert a positive test leap second. Real leap seconds are applied at 23:59:60 (UTC) as shown below.

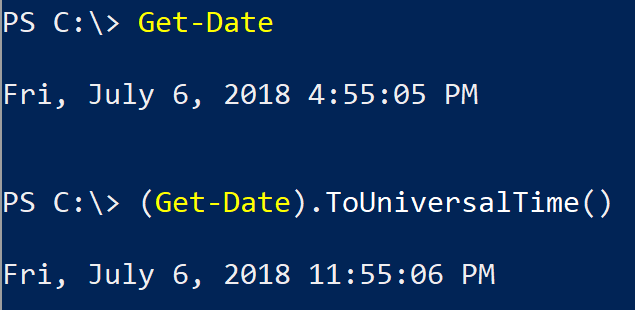
UTC-Timeline during a leap second

|  |  |
| --- | --- |
| **Without a Leap Second** | **With a Leap Second** |
| 23:59:58 | 23:59:58 |
| 23:59:59 | 23:59:59 |
| **00:00:00** | **23:59:60** |
| 00:00:01 | 00:00:00 |
| 00:00:02 | 00:00:01 |

The taskbar clock displays local time which is specific to your time zone. For example, in PST, the clock would look like this during a leap second (24-hour notation is used here for simplicity).

|  |  |
| --- | --- |
| **Without a Leap Second** | **With a Leap Second** |
| 16:59:58 | 16:59:58 |
| 16:59:59 | 16:59:59 |
| **17:00:00** | **16:59:60** |
| 17:00:01 | 17:00:00 |
| 17:00:02 | 17:00:01 |

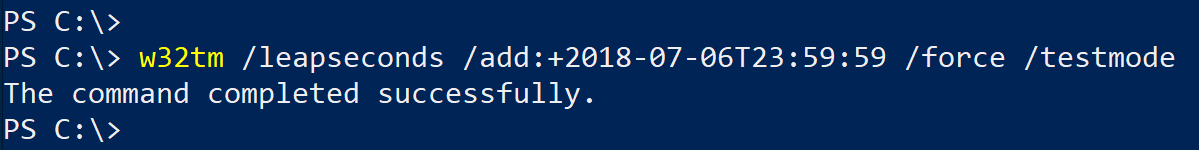
For our test, we will insert test leap-seconds; the commands accept UTC time entries. To identify the UTC time, you could change your time zone to reflect UTC, or use the following commands to compare local time with UTC time.



The UTC date and time in the above picture is July 6th, 11:55:06 PM or 23:55:06 in 24-hour format.

Next insert a positive leap second using the command:

**w32tm /leapseconds /add:+2018-07-06T23:59:59 /force /testmode**

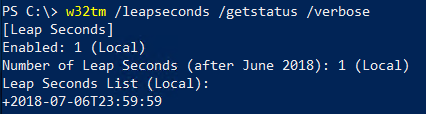


**Note**: To perform multiple tests, you can use the options in the [**Before you begin**](#_Before_You_Begin) section to identify methods to change the date and time.

Next, use w32tm – an inbox tool for managing the time service – to view and verify the leap second configuration.

To view the existing leap seconds on the system, run the following command:

**w32tm /LeapSeconds /GetStatus /Verbose**

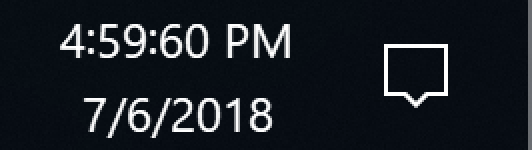


## Activity 4: Observe the leap second

In this activity, we will observe the leap second just inserted. The simplest way to do this, is to watch the task bar which was configured to display seconds in a previous activity. We will also observe event logs and file system modified and creation times.

As noted previously, the taskbar clock displays local time but the leap second is entered in UTC time. Review the [**Before you begin**](#_Before_You_Begin) section to convert the time and verify the taskbar is displaying the local time which reflects the UTC time at which the leap second in inserted.

For example, the test leap second above was entered for July 6th, 2018 at the end of the 23rd hour. In PST (+7 as, at the time of writing we’re in a DST shift), the taskbar should reflect 4:59:60 during a leap second.



Next, using PowerShell, you can automate the creation of event logs and files and observe the output. Run the following PowerShell commands as the time rolls towards the leap second.

$logFilePath = 'C:\Temp\LeapSecondFiles'

New-Item -Path $logFilePath -Type Directory -Force

while ($true) {

$date = Get-Date

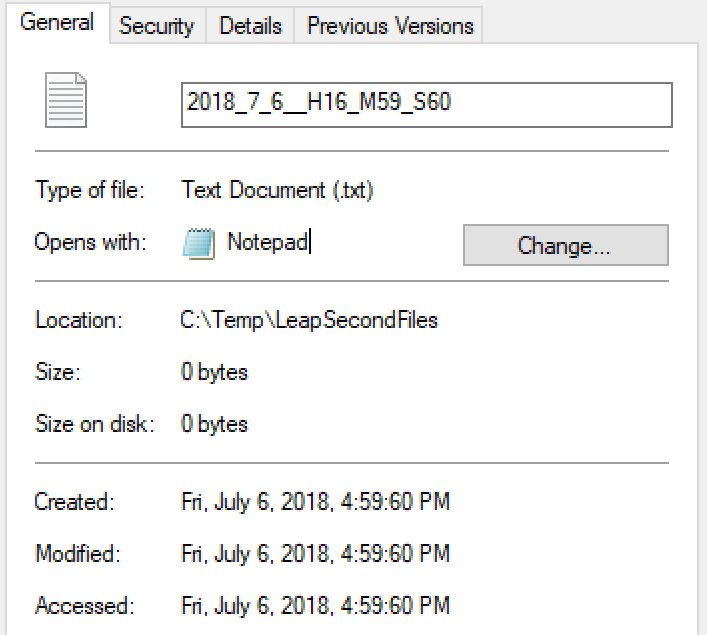
New-Item -ItemType File -Path $logFilePath `

-Name "$($date.Year)\_$($date.Month)\_$($date.Day)\_\_H$($date.Hour)\_M$($date.Minute)\_S$($date.Second).txt" -Force

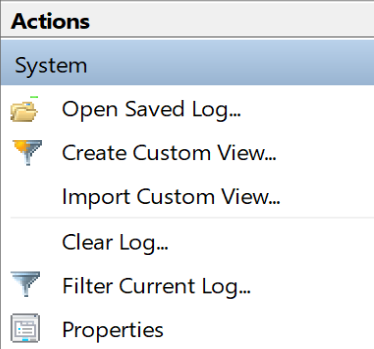
eventcreate.exe /L System /D 'Leap Second Test' /T Error /ID 999

}

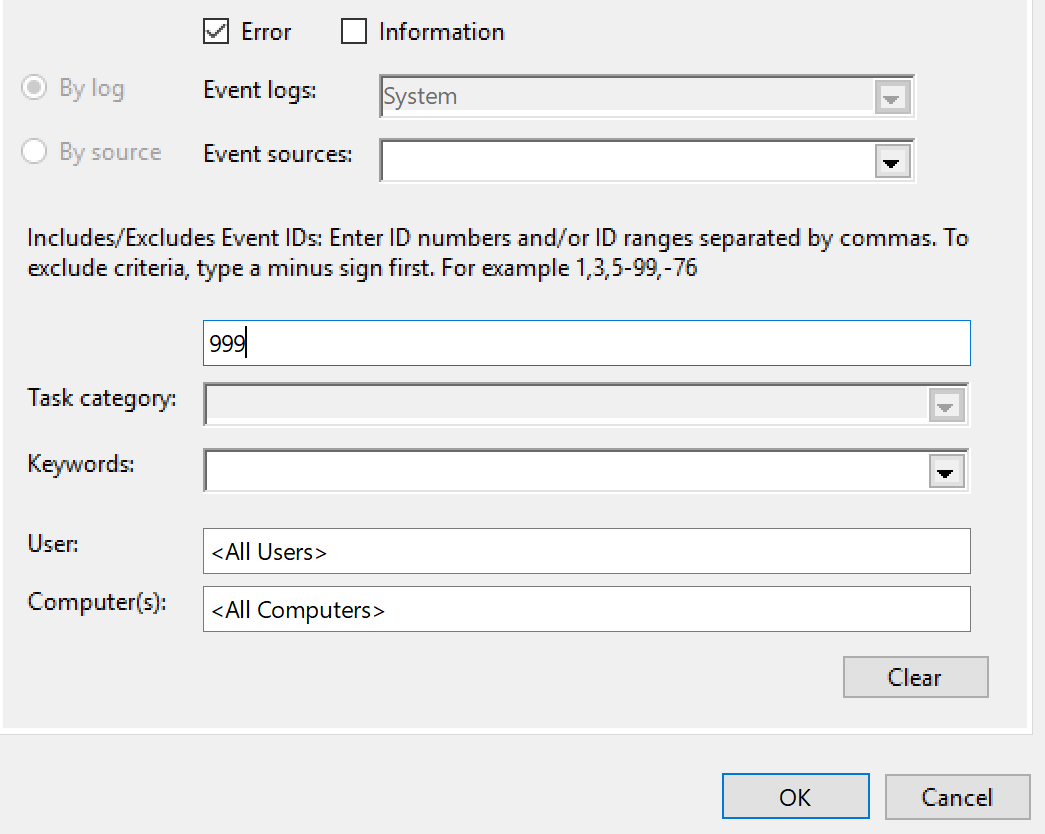
Next review the files created



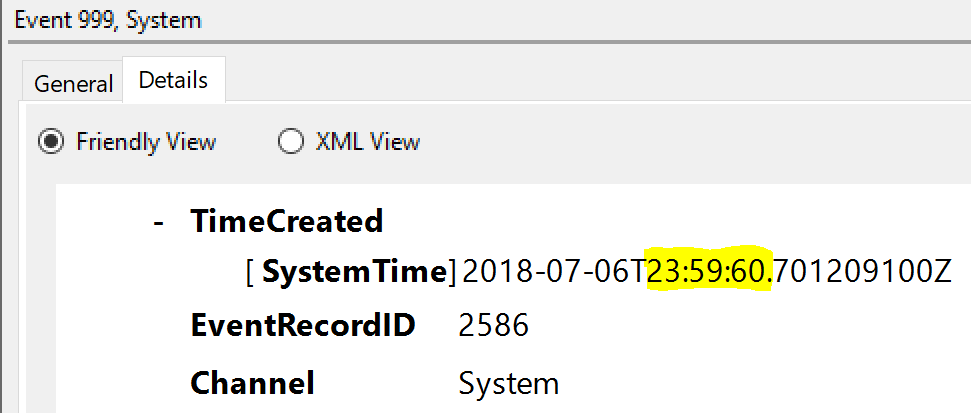
Now open the System event log and click the **Filter Current Log** option:



Select to filter **Error** and event **999**. Now click **OK**.



Select one of the events and review the **Details** tab **Friendly View**.



## Activity 5: Insert Negative Leap Seconds

Although one has never occurred, negative leap seconds could also be inserted into UTC. A negative leap second is where one second is removed from UTC.

UTC-Timeline during a negative leap second

|  |  |
| --- | --- |
| **Without a Leap Second** | **With a Leap Second** |
| 23:59:58 | 23:59:58 |
| 23:59:59 | **00:00:00** |
| **00:00:00** | 00:00:01 |
| 00:00:01 | 00:00:02 |
| 00:00:02 | 00:00:03 |

Next insert a negative leap second using the command:

**w32tm /leapseconds /add:-2018-07-06T22:59:59 /force /testmode**

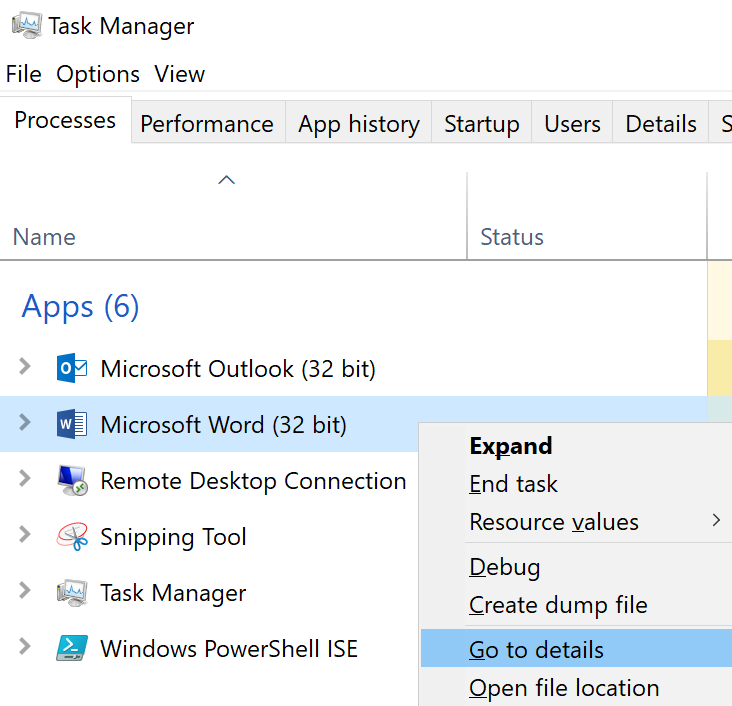
**Note:** In this example we’ve added a negative leap second on the hour before our previous test of the positive leap second. You may need to reboot to see this applied properly.

## Activity 6: Test an Application

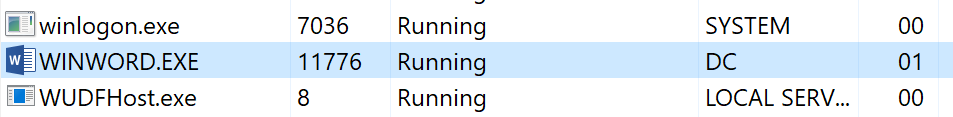
Applications must be written to consume and process leap seconds – By default, we assume that applications are not leap-second aware. As an IT Pro, we anticipate that you will want to test and verify your application or system images during a leap second. This activity will show you how you can manually opt-in an application and test its functionality.

**Note**: If you are a developer looking to make your application leap second aware, please visit <https://aka.ms/Dev-LeapSecond>. Alternatively, if you are IT Pro whose third-party application currently does not support leap seconds, please contact your vendor and point them to <https://aka.ms/Dev-LeapSecond> for guidance.

To opt-in an application, identify the process name of the application. This is easily done using task manager. Right-click on the application you wish to opt-in and select **Go to Details**



Note the highlighted process:

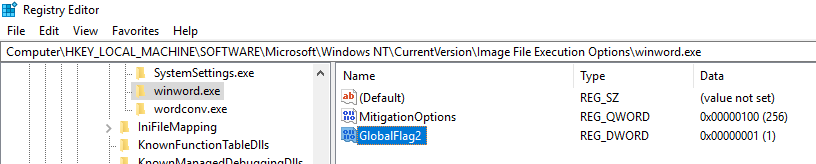


Next open the registry editor and navigate to

**HKLM:\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options**

Add a key which is the same name as the process you want to opt-in to leap seconds. In this example, we’ve opted-in the **winword.exe** process by creating a Registry Key (folder icon).

Next create a REG\_DWORD named **GlobalFlag2** with a value of **1** as shown here.



The application and processes must be re-launched to take effect.

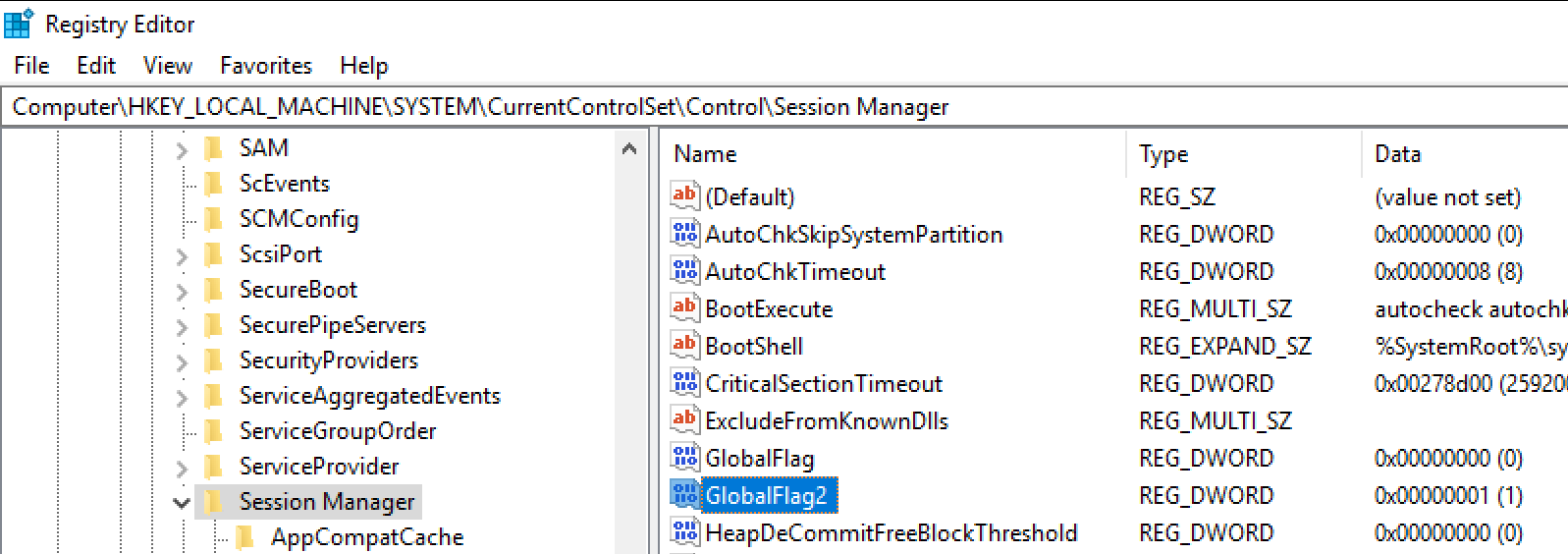
Now insert leap seconds as before and test critical application functionality.

## Activity 7: Test a System

Instead of testing an individual application one-by-one, you may want to test a holistic system. To do this, open the registry editor and navigate to

**HKLM:\ SYSTEM\ControlSet001\Control\Session Manager**

Next create a REG\_DWORD named **GlobalFlag2** with a value of **1** as shown here.



Restart the system for this to take effect.

Now insert leap seconds as before and test critical application functionality.

The following script can be used to insert positive and negative leap seconds at the end of each hour till the end of the current UTC day.

$UTCDate = (Get-Date).ToUniversalTime()

$UTCDate.Hour..23 | ForEach-Object {

$lsTime = "$($UTCdate.Year)-$($UTCdate.Month)-$($UTCdate.Day)T$($\_):59:59"

if($\_ % 2 -eq 0 ) {

w32tm.exe /leapseconds /add:+$lsTime /testmode /force

}

Elseif($\_ % 2 -eq 1 ) {

w32tm.exe /leapseconds /add:-$lsTime /testmode /force

}

}