TimeShiftX® 7.x

Windows Manual

Table of Contents

The rest of this page is left intentionally blank.

Overview

What is TimeShiftX (TSX)? How does it work?

TimeShiftX (TSX) is a simulation software that lets you time travel software into the future or past for temporal testing of all date sensitive functionality and code. TSX uses virtual clocks to enable time travel inside of Active Directory & Kerberos without changing system clocks, changing code, or isolating servers. TSX works by intercepting the OS date & time calls made by software programs and swapping them with your custom user-created virtual time, all the while leaving your OS system files, timestamps and security tokens untouched.

System Requirements

Operating System	Windows Server® x86/x64, Windows® x86/x64
Available Disk Space	60 MB
Memory	64 MB RAM
Access rights	Local administrator rights
Software	.NET Framework 4.0 or greater
	Microsoft Visual C++ 2010 Package x64, x86

Note: For Windows Server Core, see below .NET 4.0 download link & install steps
http://www.microsoft.com/en-us/download/details.aspx?id=22833

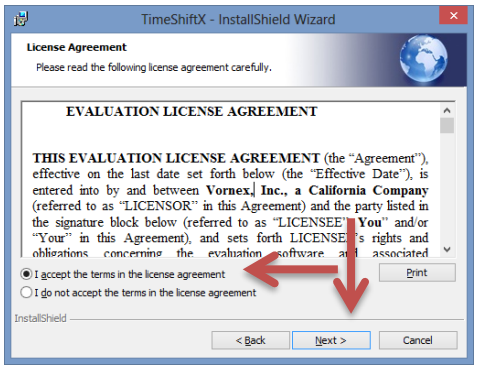
Technical Support

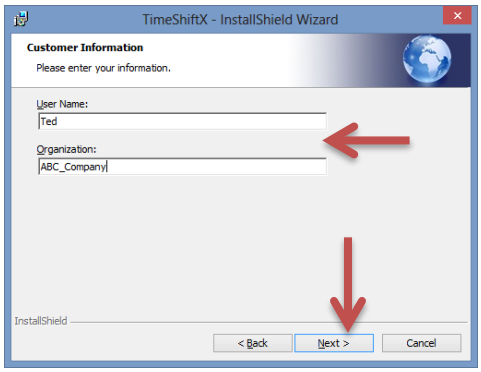
For assistance, email support@vornexinc.com or call 408.713.1400 ext. 2

Install TSX

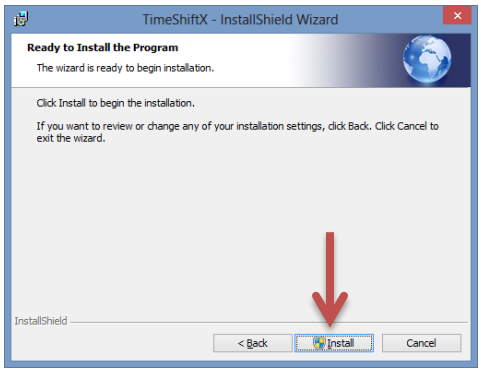
Note: TSX installs onto the primary OS partition (typically C:\) as is required.

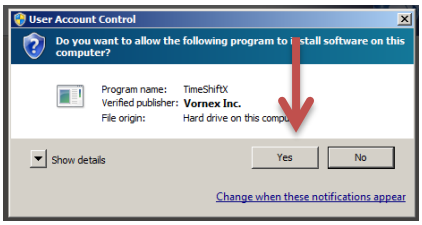
1) Run the TSX msi installer with admin rights.
2) Click **Next**
3) Select the top radial button **“I accept the terms in the license agreement”**.
4) Click **Next**

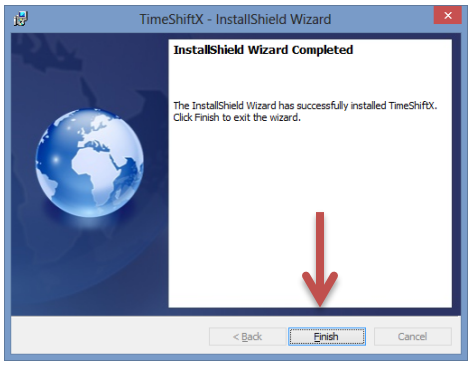




5) Enter your user and organization
6) Click **Next**
7) Click **Install**







8) Click **Yes** if a “User Access Control” dialogue appears
9) Click **Finish**

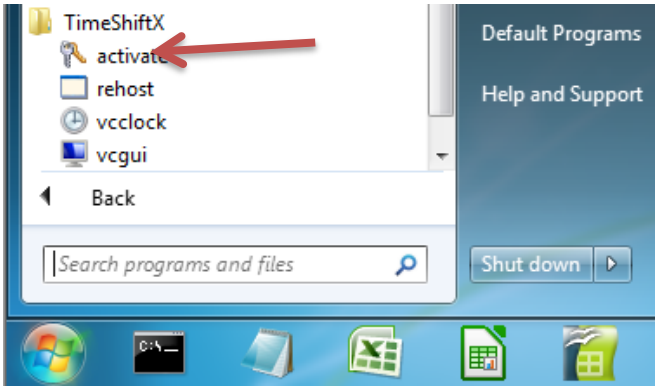
Install complete. An OS restart is **not** needed.

Now License TSX

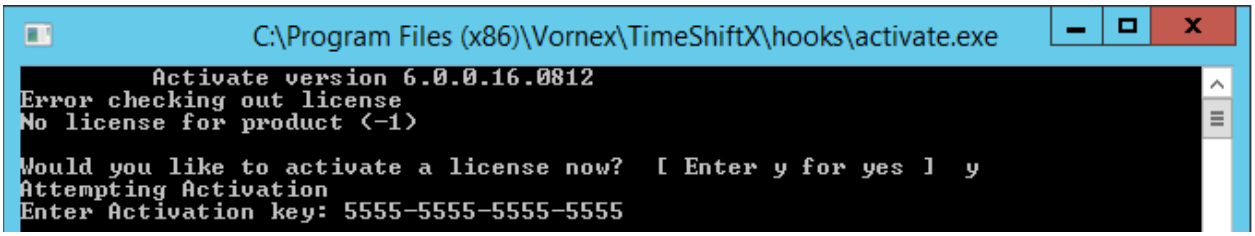
License TSX

Your server needs internet access. If none, skip to step 4 a)
Below is for trial & node-locked licenses. For floating licenses, email support@vornexinc.com
Note: The license utility will connect to http://hostedactivation.com/ on port 80.

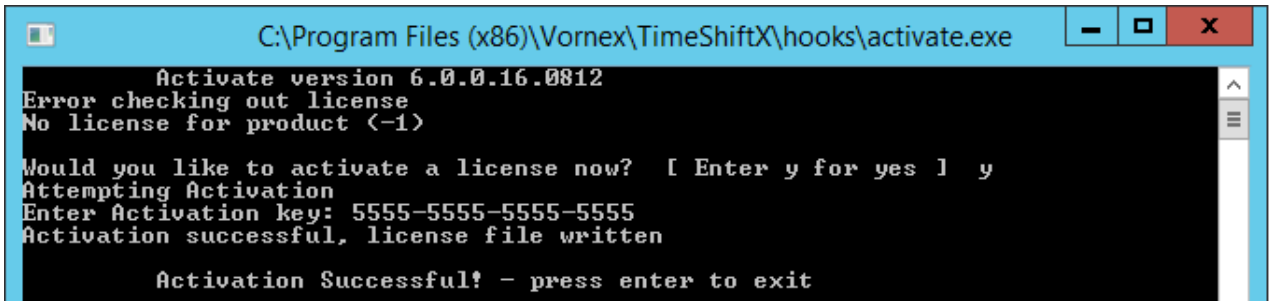
1) Click **Start** -> **Program Files** -> **TimeShiftX** -> **activate**



2) Type **Y** then press **Enter**
3) Type your 16-digit key then press **Enter**



4) If successful, you will see "Activation Successful". Press **Enter** to finish.
   a. If unsuccessful, send the server output of `ipconfig /all` to support@vornexinc.com
   b. Optionally: If behind a firewall, open port 80 and retry key.

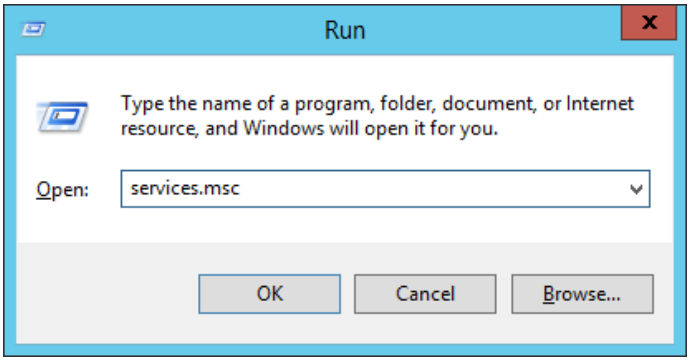


Note: Do not change the OS system clock as it will break the license.

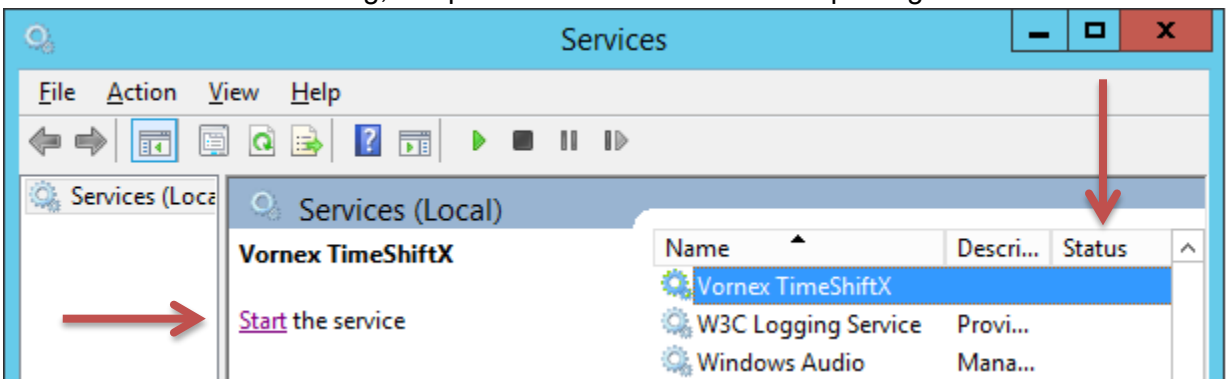
Licensing complete. Now Turn On TSX

Turn On TSX

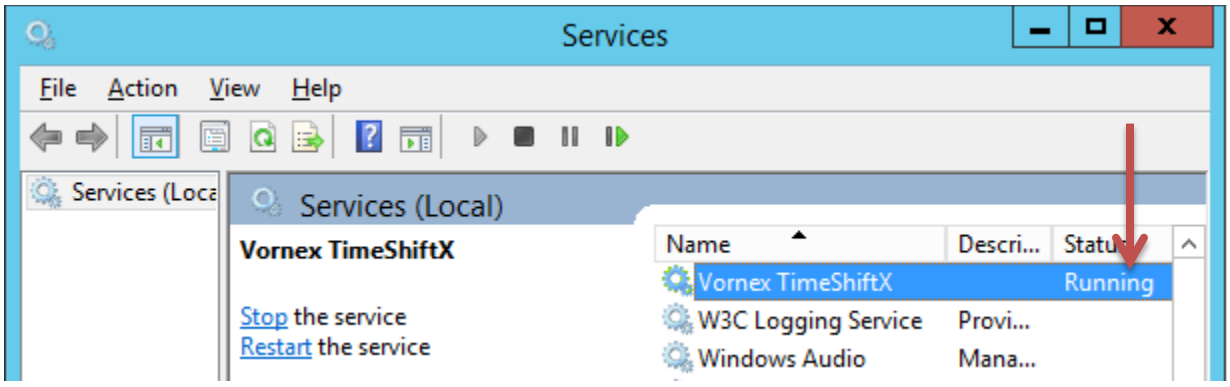
1) Click **Start** -> Type **Run** -> Press **Enter** -> Type **services.msc** -> Click **OK**



2) Click **Vornex TimeShiftX**
3) Click **Start**
    a. If “Start” is missing, re-open services.msc with admin privileges.



4) After, it will show **Running** or **Started** in the Status column.
    a. If service fails to start, re-run “activate” and send screenshot to support@vornexinc.com
    b. TSX can take up to 15 seconds to fully initialize.
    c. TSX service is set for “Local System” in the “Log on as” field. Do NOT change this.



TSX is now running. Begin time travelling!

Use TSX - GUI

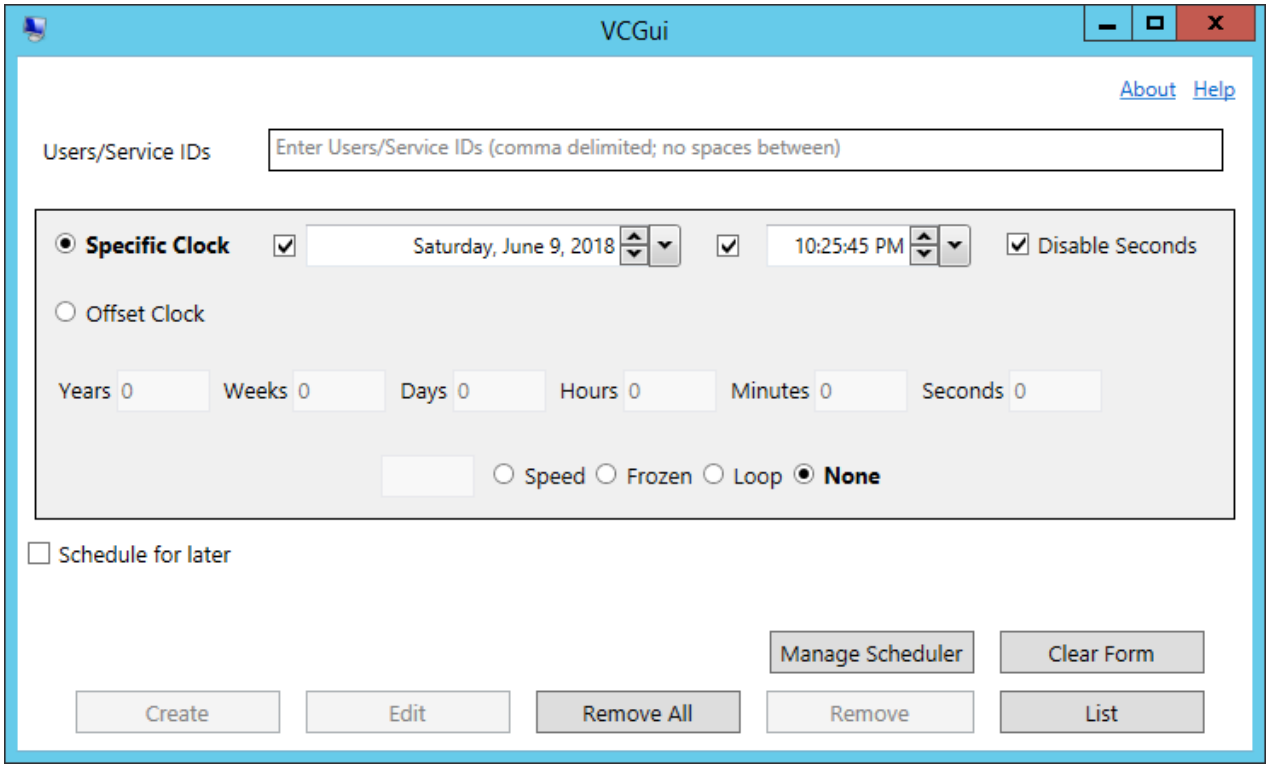
1) Click **Start** -> **Program Files** -> **TimeShiftX** -> **vcgui**

2) Enter the user or service accounts (without domain info) that run the apps you want to time travel
   a. Find user or service accounts using the Task Manager “Details” tab or Services.msc
   b. Multiple users are comma delimited with no spaces between
   c. To time travel PIDs, edit line 23 of the below file to be “true” and re-open the GUI.
      `C:\Program Files (x86)\Vornex\TimeShiftX\VCGui.exe.config`
      → `<add key="PID" value="True"/>`

3) Select **Specific** or **Offset** radial buttons and enter your desired time values

4) Click **Create**
   a. After, all apps running as the users will immediately time travel to your defined virtual time.
   b. Apps are any binary / exe on the system, whether a database, java application, etc.
   c. To pick & choose which exact apps to time travel only see Include or Exclude Apps

5) When finished time travelling, click **Remove All** to revert all users & apps to present system time.



GUI Button Descriptions

Specific Clock	Virtual times created by defining an exact date and time. Select a date and time using the drop-down widgets. You can set only a date, only a time, or both via the checkboxes.
Offset Clock	Virtual times created by adding or subtracting a value relative to the real system time. Enter integers in the 6 fields. Unused fields must list ‘ 0 ‘ Positive integers go to the future; negative integers, with a hyphen, go to the past.
Create	Creates virtual times
Edit	Edits existing virtual times by allowing you to add or subtract values or adjust the speed or type of an existing virtual time. Enter any Offset Clock Value and-or set a Speed / Frozen / Loop option.
Remove All	Deletes all virtual times. All users and apps will revert back to present system time.
Remove	Deletes individual virtual times. Only the users/PIDs defined in the top field, and their respective apps, will revert back to present system time.
List	Lists all virtual times
Clear Form	Clears all entered values
Manage Scheduler / Schedule for Later	Read Use TSX GUI - Scheduler
Disable Seconds	If checked, TSX will not time travel seconds but will use the system seconds when creating Specific virtual times. We recommend to checkbox this field to **not** time travel seconds so that your servers stay in sync with your NTP.
None	Causes the virtual time to progress forward as a normal clock.
Loop	Creates a virtual time that will tick forward normally to 23:59:59 (11:59:59 PM) then will roll back over to 00:00:00 of the same day, thus always staying on the same day. • A looping virtual clock can only be set to loop on one single day. • To progress forward but keep your looping virtual clock, use the Edit feature. • You cannot use speed or frozen flags with looping.
Frozen	Causes the virtual time to stop progressing forward and be frozen.
Speed	Creates a virtual time that travels N times faster or slower than real time. • A speed value equal to the number 1 is real time speed • A speed value greater than the number 1 is faster than real time • A speed value less than the number 1 is slower than real time • Speed value notation is 5 digits to the left and right of the decimal point. {XXXXX.nnnnn} • Speed is in relation to how fast seconds travel. See below examples

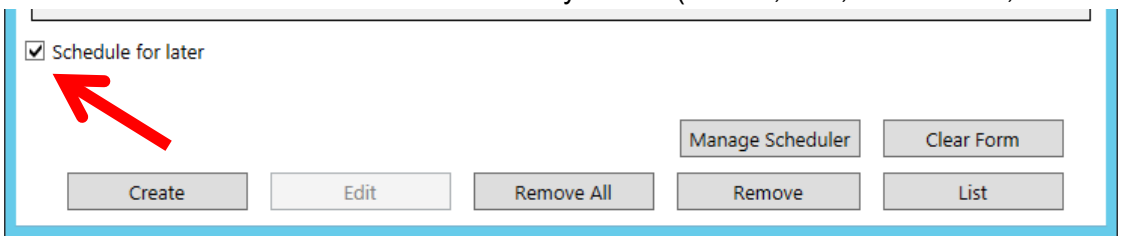
Normal Speed	-v 1
2 times faster	-v 2
4 times faster	-v 4
10 times faster	-v 10
2 times slower	-v 0.5
4 times slower	-v 0.25
10 times slower	-v 0.1

Travel 1 minute every second	-v 60
Travel 10 minutes every second	-v 600
Travel 30 minutes every second	-v 1800
Travel 1 hour every second	-v 3600
Travel 1 day every second	-v 86400
Travel 1 second every minute	-v 0.016
Travel 1 second every 10 minutes	-v 0.00166

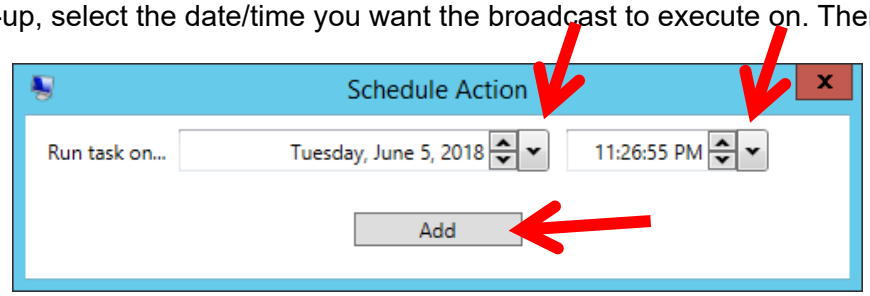
Use TSX GUI - Scheduler

To schedule commands to be run at a later time, use the Scheduler feature via the below. Note: vcgui must be open or minimized for scheduled tasks to work. If closed, all scheduled tasks will be deleted.

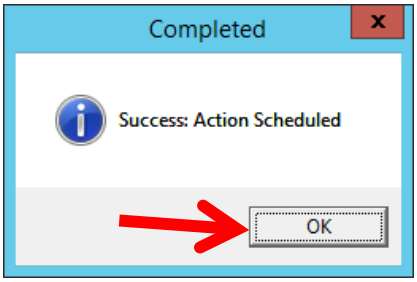
1. Check the "**Schedule for later**" box and click any button (Create, Edit, Remove All, or Remove).



2. In the pop-up, select the date/time you want the broadcast to execute on. Then click **Add**

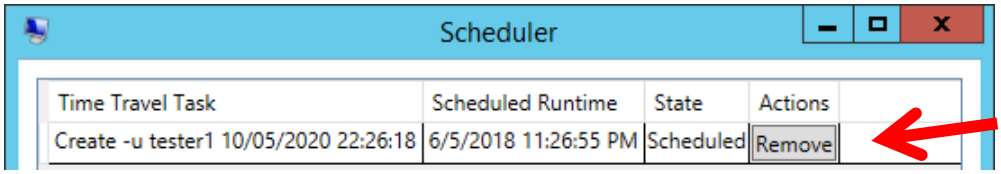


3. Click **OK** on the success message.

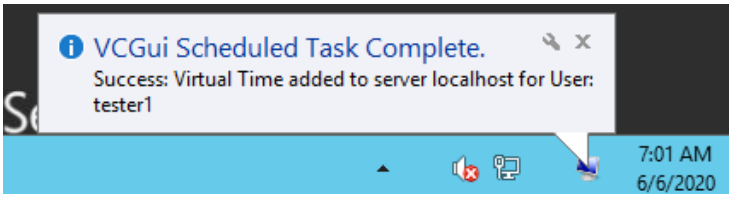


Done! Now your time travel action is scheduled to be run at a later date.

To view your scheduled tasks click the "**Manage Scheduler**" button to reveal the below window. To remove scheduled tasks click "**Remove**" on the far right.



Once a task completes, you will see the results the file "Scheduler_Results.txt" (inside the TSX log folder) and an OS system tray message, like below.



Use TSX - Terminal

Use the tsx CLI terminal application to create/edit/remove/list a virtual clock.

There are two types of Virtual clocks: USER and PID. See below chart.

USER	**Time travel by OS user / service account** **Most popular option** E.g. `C:\> tsx set -u user1 -y 1` When you time travel an OS user / service account, then all or your selected apps running as that user will immediately time travel. (Pick & choose apps via Include or Exclude Apps)
PID	**Time travel by process id (PID)** E.g. `C:\> tsx set -p 3453 -y 2` When you time travel a PID, then the apps running as that PID will immediately see the virtual clock. Note • If the app restarts the PID # will change, thus you will need to re-run TSX with the new PID # • If the app spawns new PIDs you will need to run a new TSX command with the new PIDs. • Once a time traveling PID terminates, TSX will automatically disable the time travel for that terminated PID # as it has been released back to the OS to be used for another process.

Virtual clocks will progress forward in time (as a normal clock) unless otherwise defined.

User accounts are entered without domain information. Accounts with special symbols or multiple spaced out words must be wrapped in double quotes.

Multiple user accounts or PIDs are space delimited.

When receiving virtual times, the precedence of application is PID, then USER virtual times.

Create an Offset Clock

```
C:\> tsx set -u {Account Names} [-y | -w | -d | -h | -m | -s] {integer}

C:\> tsx set -p {PID Numbers} [-y | -w| -d | -h | -m | -s] {integer}
```

Offset clocks are virtual times created by adding or subtracting a value relative to the real system time. Use any combination of the below flags. Positive integers go in the future; negative integers, with a hyphen, go in the past.

-y {integer}	Years	-h {integer}	Hours
-w {integer}	Weeks	-m {integer}	Minutes
-d {integer}	Days	-s {seconds}	Seconds

Examples	
Time travel Joe and Tom 5 years and 2 weeks in the *future*	`C:\> tsx set -u Joe Tom -y 5 -w 2`
Time travel user Oracle for 4 weeks and 10 days in the *past*	`C:\> tsx set -u oracle -w -4 -d -10`
Time travel user "UAT JbossA" 3 hours in the *future*	`C:\> tsx set -u "UAT JbossA" -h 3`
Time travel PID 123 and 234 1 year into the *future*	`C:\> tsx set -p 123 456 -y 1`

Create a Specific Clock

```
C:\> tsx set -u {Account Names} [mm/dd/yyyy hh:mm:ss]

C:\> tsx set -p {PID Numbers} [mm/dd/yyyy hh:mm:ss]
```

Specific clocks are virtual times created by defining an exact date and time you want to jump to. Allowed date/time formats are below

mm/dd/yyyy hh:mm:ss	Date and time
mm/dd/yyyy hh:mm	Date and time (without seconds)
mm/dd/yyyy	Date only. 2-digit month / 2-digit day / 4-digit year
hh:mm:ss	Time only. 2-digit hour : 2-digit minute : 2-digit seconds
hh:mm	Time only (without seconds). 2-digit hour : 2-digit minute

Note:

- Dates are in US Standard format
- Times are in 24-hour format
- If you only define the date, the current system time will be used.
- If you only define the time, the current system date will be used
- We recommend to **not** time travel seconds so that your servers stay in sync with your NTP

Examples	
Time travel user Ora1 to July 11th 2025 11:45 PM	`C:\> tsx set -u ora1 07/11/2025 23:45`
Time travel users sql1 and Joe for July 30th 2006	`C:\> tsx set -u sql1 joe 07/30/2006`
Time travel PID 789 to June 10th 2010	`$ tsx set -p 789 06/10/2010`

Delete all virtual times

```
C:\> tsx reset
```

All apps will revert back to present time.

Delete individual virtual times

```
C:\> tsx remove -u {Account Names}

C:\> tsx remove -p {PID Numbers}
```

Only apps running as the defined user accounts or PIDs will revert back to system time.

List virtual times

```
C:\> tsx list
```

Note: The below Freeze, Speed, Loop, & Edit commands apply to USER and PID virtual times. Just replace "-u {Account Names}" with "-p {PID Numbers}"

Freeze virtual times

```
C:\> tsx set -u {Account Names} [-y |-w | -d | -h | -m | -s] {integer} -f

C:\> tsx set -u {Account Names} [mm/dd/yyyy hh:mm:ss] -f
```

Append the ' –f ' flag to a virtual time to cause it to stop progressing forward and be frozen.

Speed virtual times faster or slower

```
C:\> tsx set -u {Account Names} [-y | -w | -d | -h | -m | -s] {integer} -v {integer}

C:\> tsx set -u {Account Names} [mm/dd/yyyy hh:mm:ss] -v {integer}
```

Append the ' -v ' flag to a virtual time followed by an integer to have your virtual times travel N times faster or slower than real system time.

- A speed value equal to the number 1 is real time speed
- A speed value greater than the number 1 is faster than real time
- A speed value less than the number 1 is slower than real time
- Speed value notation is 5 digits to the left and right of the decimal point. {XXXXX.nnnnn}

Speed is in relation to how fast seconds travel. See below examples

Normal Speed	-v 1
2 times faster	-v 2
4 times faster	-v 4
10 times faster	-v 10
2 times slower	-v 0.5
4 times slower	-v 0.25
10 times slower	-v 0.1

Travel 1 minute every second	-v 60
Travel 10 minutes every second	-v 600
Travel 30 minutes every second	-v 1800
Travel 1 hour every second	-v 3600
Travel 1 day every second	-v 86400
Travel 1 second every minute	-v 0.016
Travel 1 second every 10 minutes	-v 0.00166

Examples	
Time travel Tom 5 years in the future at a speed 5x faster than real time	`C:\> tsx set -u Tom -y 5 -v 5`
Time travel Bob to July 10th 2010 at a speed of 4x slower than real time	`C:\> tsx set -u Bob 07/10/2010 -v .25`

Loop virtual times

```
C:\> tsx set -u {Account Names} [-y | -w | -d | -h | -m | -s] {integer} -l

C:\> tsx set -u {Account Names} [mm/dd/yyyy hh:mm:ss] -l
```

Append the ' -l ' (lower case L) flag to a virtual clock to cause it to loop. This means the virtual clock will tick forward normally to 23:59:59 (11:59:59 PM) then it will roll back over to 00:00:00 of the same day, thus always staying on the same day.

Note 1: A looping virtual clock can only be set to loop on one single day.

Note 2: To progress forward but keep your looping virtual clock, use the Edit feature.

Note 3: You cannot use speed or frozen flags with looping.

Examples	
Set a virtual time for Bob to always loop on October 10th 2020	`C:\> tsx set -u Bob 10/10/2020 -l`
Create a looping virtual clock 1 year and 10 days into the future for Tom and Steve	`C:\> tsx set -u Tom Steve -y 1 -d 10 -l`
Travel Larry to July 2 2002 at 10AM and loop it on that day	`C:\> tsx set -u Larry 07/02/2002 10:00 -l`

Edit existing virtual times

```
C:\> tsx set -e -u {Account Names} {-y | -w | -d | -h | -m | -s} {integer}
```

Append the '-e' flag after ' set ' and before ' -u ' and add any of the below combinations of flags to add or subtract values to an existing virtual time or to adjust its speed. Positive integers go in the future; negative integers, with a hyphen, go in the past. The existing virtual time type (normal, frozen, speed, loop) will be preserved unless defined in the new edit command in which case the type will be changed.

-y {integer}	Years	-h {integer}	Hours
-w {integer}	Weeks	-m {integer}	Minutes
-d {integer}	Days	-s {integer}	Seconds
-f	Freeze speed*	-v {integer}	Speed value*
-l	Loop*		

*Refer to speed, frozen, looping sections in manual

Examples	
Add 1 week (into the future) to Phil's current virtual time.	`C:\> tsx set -e -u Phil -w 1`
Add 10 days and 1 hour (into the future) to Tom and Steve's existing virtual time	`C:\> tsx set -e -u Tom Steve -d 10 -h 1`
Adjust Bob's virtual time to travel 10x faster than real-time.	`C:\> tsx set -e -u Bob -v 10`
Subtract 1 year (into the past) to Bill's existing virtual time and adjust it to travel 1/4 slower than real time	`C:\> tsx set -e -u Bill -y -1 -v .25`

Delete all User virtual times only

```
C:\> tsx reset users
```

All user virtual clocks will be deleted only. PID virtual clocks will remain.

Delete all PID virtual times only

```
C:\> tsx reset pids
```

All PID virtual clocks will be deleted only. User virtual clocks will remain.

List exact types of virtual times

```
C:\> tsx list users

C:\> tsx list pids
```

Listing specific virtual time field values

```
C:\> tsx list [-u|-p] {Account Names} {datetime | date | month | monthname | day | dayname | year | time | hour | minute | second}
```

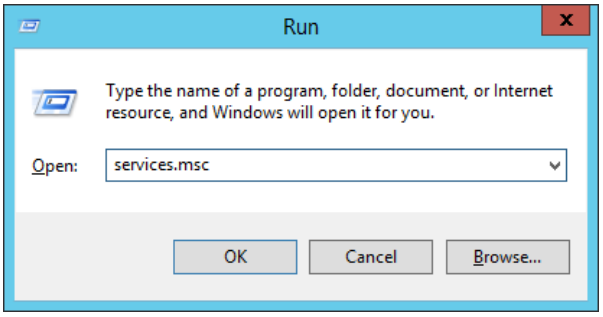
The list command allows for extraction of specific individual virtual time field values. Append the '-u' or '-p' flag after ' list ' and define the user account or PID and any one of the below parameters

datetime	date	month	monthname
day	dayname	year	time
hour	minute	second	

Command Examples	Output
For these examples the user account Bob has an existing virtual time of 07/10/2020 10:30:15	
`C:\> tsx list -u bob datetime`	`C:\> 07/10/2020 10:30:15`
`C:\> tsx list -u bob date`	`C:\> 07/10/2020`
`C:\> tsx list -u bob month`	`C:\> 07`
`C:\> tsx list -u bob monthname`	`C:\> July`
`C:\> tsx list -u bob day`	`C:\> 10`
`C:\> tsx list -u bob dayname`	`C:\> Friday`
`C:\> tsx list -u bob year`	`C:\> 2020`
`C:\> tsx list -u bob time`	`C:\> 10:30:15`
`C:\> tsx list -u bob hour`	`C:\> 10`
`C:\> tsx list -u bob minute`	`C:\> 30`
`C:\> tsx list -u bob second`	`C:\> 15`

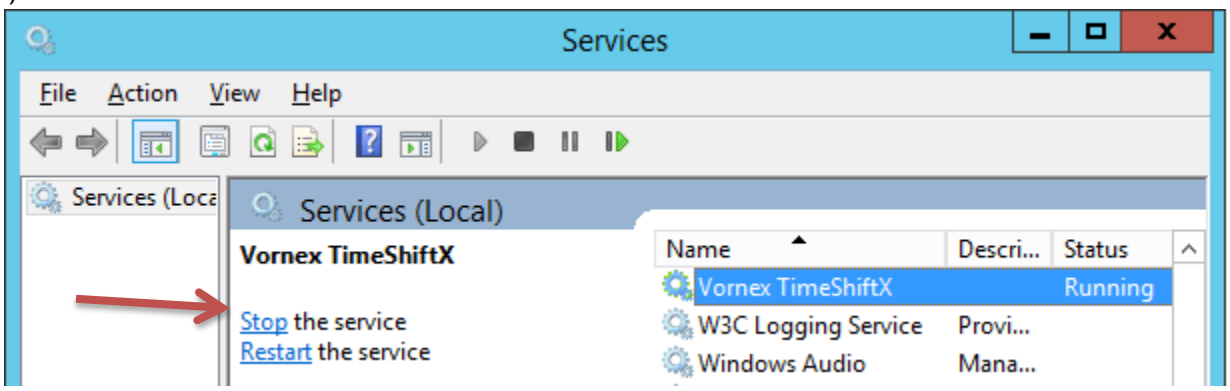
Uninstall TSX

1) Click **Start** -> Type **Run** -> Press **Enter** -> Type **services.msc** -> Click **OK**

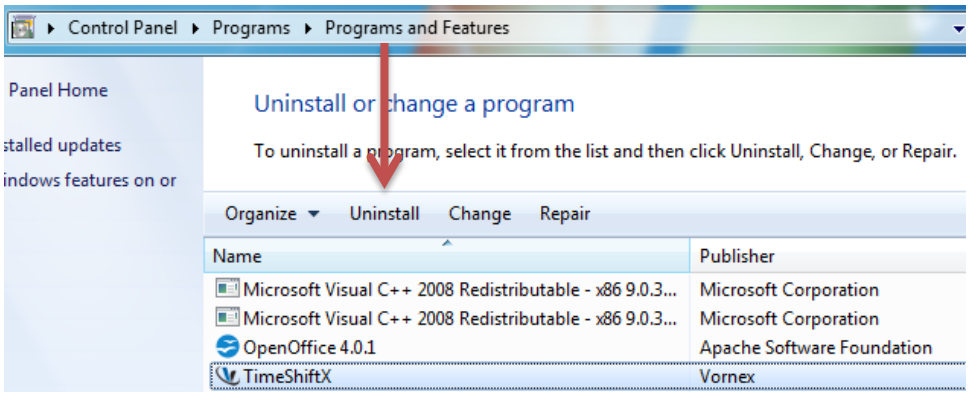


2) Click **Vornex TimeShiftX** -> Click **Stop**

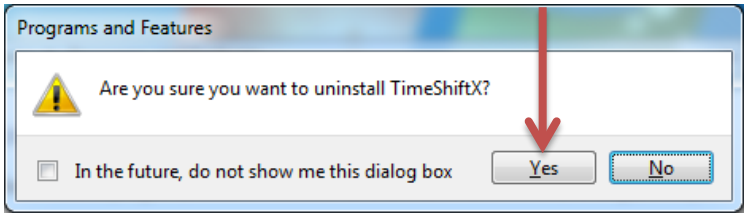
3)



1) Click **Start** -> **Control Panel** -> **Program and Features** -> **TimeShiftX** -> **Uninstall**



2) Click **Yes** to the pop-up dialogue



Done. TSX is now uninstalled.

Include or Exclude Apps (Binaries / EXE's)

By default, TSX time travels all apps for defined OS user accounts. This is the most popular setting.

To include / exclude specific apps, edit our HookingRules.JSON per the below steps and examples.

Inclusion means to only time travel defined app/users.

Exclusion means to time travel all desired apps/users except the defined apps/users.

Include or Exclude Steps

1. Turn OFF the "Vornex TimeShiftX" service in services.msc
2. Edit file `C:\Program Files (x86)\Vornex\TimeShiftX\HookingRules.JSON`
    a. On 32bit OS the location is "Program Files"
    b. Directory paths must use forward slashes ' / '
    c. The last element/object cannot have a trailing comma
3. Turn ON the "Vornex TimeShiftX" service in services.msc
    a. Formatting errors will cause the service to not start.
    b. Defined users and processes in JSON are not case sensitive.
4. Done! Follow below examples in yellow

Exclude Apps

```
{
"rules": {
      "hook": "all",
      "noHook": [
            {"user": "SYSTEM"},
            {"user": "LOCAL SERVICE"},
            {"user": "NETWORK SERVICE"},
            {
            "files": [
                  {"name": "tsx.exe"},
                  {"name": "notepad.exe"},
                  {"name": "C:/Program Files (x86)/Example/clockA.exe"},
            …
```

Steps: Under "noHook" in "files", add a "name" element and define an app.

Meaning: All apps with the defined name will not time travel. Adding the exact path means only that specific app will not time travel.

Include Apps

```
{
"rules": {
      "hook": [
      {
            "files": [
                  {"name": "notepad.exe"},
                  {"name": "C:/Program Files (x86)/Example/clockA.exe"}
                  ]
       }
      ],
      "noHook": [
            {"user": "SYSTEM"},
            {"user": "LOCAL SERVICE"},
      …
```

Steps: Replace “hook”: “all” with above template and populate “name” elements with your apps.
Meaning: All apps with the defined name will only time travel. Adding the exact path means only that specific app will time travel.

Warning: Only the defined apps will time travel. New apps/users must be manually added to JSON

Include Users

```
{
"rules": {
      "hook": [
            {"user": "tester1"},
            {"user": "tester2"}
      ],
      "noHook": [
```

Steps: Replace “hook”:”all” with above template and add “user” elements with a defined user account.
Meaning: Only apps running as the defined users will ever time travel or be injected with TSX library.
Warning: This is **not recommended** as the GUI, CLI, Broadcaster can control which users time travel.

Exclude Users

```
{
"rules": {
      "hook": "all",
      "noHook": [
            {"user": "SYSTEM"},
            {"user": "LOCAL SERVICE"},
            {"user": "NETWORK SERVICE"},
            {"user": "Bob"},
      …
```

Steps: Under “noHook”, add a “user” element with a defined user account.
Meaning: All apps running as the defined user will not time travel or be injected with TSX library.
Warning: This is **not recommended** as the GUI, CLI, Broadcaster can control which users time travel.

Include System Apps for Time Travel

By default, TSX does not time travel the system accounts `LOCAL SERVICE`, `NETWORK SERVICE`, and `SYSTEM` (aka `LOCAL SYSTEM`) as it can negatively impact file timestamps, security tokens, backups, etc. and overall system stability. We recommend apps running as non-system accounts as using the system accounts is viewed as security risk by most vendors.

However, you can target specific apps running as system accounts to time travel only.

Example: Time Travel Apps *w3wp.exe* and *clockA.exe* that run as SYSTEM

```
{
"rules": {
      "hook": [
      {"user": "tester1"},
      {
            "files": [
                  {"name": "notepad.exe"},
                  {"name": "w3wp.exe"},
                  {"name": "C:/Program Files (x86)/Example/clockA.exe"}
                  ]
       }
      ],
      "noHook": [
            {"user": "LOCAL SERVICE"},
            {"user": "NETWORK SERVICE"},
      …
```

Steps:

Replace “hook”:“all” with above template and add “name” elements with your apps running as SYSTEM

Remove `{"user": "SYSTEM"},` from “noHook”

Replace “tester1” with a regular user account (for smoke-test purposes only if desired)

Add SYSTEM inside the GUI, CLI, or Broadcaster and create a virtual time

Meaning: Only apps named w3wp.exe,clockA.exe (with that exact path), and notepad.exe will time travel, even if they run as SYSTEM. Also, all apps running as “tester1” user will time travel.

Note 1: Above steps can be done with LOCAL SERVICE or NETWORK SERVICE.

Note 2: Defining notepad.exe and an extra user is helpful for smoke-test purposes.

Warning 1: Do not put SYSTEM, LOCAL SERVICE, or NETWORK SERVICE inside the “hook” section.

Warning 2: Only the defined apps/users will time travel. New apps/users must be added to JSON.

Exclude Apps for a Specific User

```
{
"rules": {
      "hook": "all",
      "noHook": [
            {"user": "SYSTEM"},
            {"user": "LOCAL SERVICE"},
            {"user": "NETWORK SERVICE"},
            {"user": "tester1",
            "files": [
                  {"name": "notepad.exe"},
                  {"name": "C:/Program Files (x86)/Example/clockA.exe"}
                  ]
            },
            {
            "files": [
                  {"name": "tsx.exe"},
            …
```

Steps: Under “noHook” add the above template and replace “tester1” with your user and add “name” elements with the apps you want to exclude for the user.

Meaning: All apps with the defined name while running as the defined user, will not time travel. Adding the exact path means only that specific binary while running as the defined user, will not time travel.

Include Apps for a Specific User

```
{
"rules": {
      "hook": [
      {"user": "tester1",
            "files": [
                  {"name": "notepad.exe"},
                  {"name": "C:/Program Files (x86)/Example/clockA.exe"}
                  ]
       }
      ],
      "noHook": [
            {"user": "SYSTEM"},
```

Steps: Replace “hook”: “all” with above template and replace “tester1” with your user and add “name” elements with the apps you want to include for the user.

Meaning: All apps with the defined name while running as the defined user, will only time travel. Adding the exact path means only that specific binary while running as the defined user, will time travel.

Warning: Only the defined apps running as that user will time travel. New apps/users must be manually added to JSON.

Chromium Browsers with JavaScript

To time travel a web app running on a Chromium based browser (Google Chrome, Microsoft Edge, Opera) that uses JavaScript date/time fields, you must enable a JSON setting via the below steps.

1. Close your browser
2. Turn OFF the “Vornex TimeShiftX” service in services.msc
3. Edit the below file and set the yellow field to “on”
   `C:\Program Files (x86)\Vornex\TimeShiftX\HookingRules.JSON`

```
"config": {
   "loggingLevel": "low",
   "pollingFallback": false,
   "lowintegrity": "on"
}
```

4. Turn ON the “Vornex TimeShiftX” service in services.msc
5. Open your browser
6. Done!

Note

- The Chromium JavaScript Engine has an internal “refresh rate” when it queries the OS date. This rate is usually every minute, thus it may take up to 60 seconds for your virtual time to appear.

- Once you start the “Vornex TimeShiftX” service, with Chrome open, you should not stop it. However, if you stop/start the service multiple times while Chrome is open, this will cause the virtual time to be added multiple times and show the wrong value in Chrome JavaScript. This is because Chrome is a special “low-integrity” process, no other process has this affect. If this happens simply restart Chrome to get the correct virtual time value.

Release Notes

For details about various items (shared apps, time zones, timers, synchronization, daylight savings, clusters / load balanced environments, Docker containers, cron, DB2, HANA, SAP, Task Scheduler, IIS, etc.) please refer to **TimeShiftX_Notes.pdf**

Silently Install & License TSX

Using the **command prompt**, follow the below in an Administrator cmd shell

1. Silent Install (need admin rights)

```
C:\> TimeShiftX.msi /quiet
```

2. Silent License (requires internet access, if none contact support@vornexinc.com)

```
C:\> cd C:\Program Files (x86)\Vornex\TimeShiftX\hooks
C:\> activate.exe key 5555-5555-5555-5555
     Error checking out license
     No license for product (-1)
     Activation successful, license file written
```

3. Turn on the TSX Service

```
C:\> sc start "Vornex TimeShiftX"
```

4. Begin testing!

Using **PowerShell**, follow the below in an Administrator PowerShell shell

1. Silent Install (need admin rights)

```
PS C:\> .\TimeShiftX.msi /quiet
```

2. Silent License (requires internet access, if none contact support@vornexinc.com)

```
PS C:\> cd "C:\Program Files (x86)\Vornex\TimeShiftX\hooks"
PS C:\> .\activate.exe key 5555-5555-5555-5555
     Error checking out license
     No license for product (-1)
     Activation successful, license file written
```

3. Turn on the TSX Service

```
PS C:\> Start-Service -Name "Vornex TimeShiftX"
```

4. Begin testing!

Silently Uninstall TSX

Using the **command prompt**, follow the below in an Administrator cmd shell

1. Silent Uninstall (need admin rights)

```
C:\> msiexec /x TimeShiftX.msi /quiet
```

Note: You can broadcast the above commands using an IT management tool like Puppet or SCCM.

Frequently Asked Questions

1. Does TSX affect system files, system clock, or file timestamps?
   No, TSX does not time travel OS File Timestamps like “Date modified, File Creation Date, etc.” as it will cause system instability. However, if an app like WebSphere generates log files with time labeled filenames or timestamp messages printed inside the logs, then those will be time travelled.

2. Does TSX work with Active Directory?
   Yes. TSX maintains all environment security tokens at all times, so you can safely time-travel your apps while inside an Active Directory or Kerberos environment.

3. Can I use TSX to run multiple date/time tests on the same server concurrently?
   Yes. This requires you to create separate instances/installs of your applications & databases under a different user account. Then you can wrap the new user accounts and the old user accounts with different virtual times.

4. What happens if the system clock is changed manually after setting a virtual time with TSX?
   The license will become invalid.