# Timezone Sync Service

August 2012  
Don Buckley ([donb@silversea.com](mailto:donb@silversea.com))

## Overview

The Timezone Sync Service (TSS) is a windows service running under Local System credentials.

On a repeating schedule that is specified by the administrator, the service checks an XML schedule file in a shared network location in order to determine what time zone the local system is supposed to be set to, and if necessary updates the time zone on the local system.

During operation, an exceptions are noted in the Windows Application log on the local system. It is important to check this log for any troubleshooting purposes.

## Prerequisites

* Windows XP / 2003 / 7 / 2008
* DotNET Framework 2.0; DotNET Framework 3.5
* 1MB Hard Disk Space; 16-32MB Memory (RAM)
* Windows Application Log set to overwrite as needed or set to adequate size if verbose logging enabled (see Registry Settings)
* Read-only network share that is accessible by the Local System account on the computer where the service is being installed.

## Installed Components

#### Files

* TimezoneSync.exe – executable for service
* Timezonesync.exe.config – (not used)
* TimezoneSync.InstallState – used by DotNET Framework for installation / uninstallation

#### Installation path

##### Windows XP / Server 2003

C:\Program Files\SilverSea\Timezone Sync Service

##### Windows 7 / Server 2008

32 bit systems: C:\Program Files\SilverSea\Timezone Sync Service  
64 bit systems: C:\Program Files (x86)\SilverSea\Timezone Sync Service

## Registry Settings

The path where the registry settings are stored is HKEY\_LOCAL\_MACHINE\Software\SilverSea\Timezone Sync Service. The 4 settings are detailed as below. The manual installation of the service will configure these settings during the install wizard.

#### XMLPath

A string value. This value is the path to the UNC read-only network share that contains the **TimezoneSchedule.xml** file. See following Operations section for details on this file.

Available for both the manual and command line installation

#### FrequencyMinutes

Integer value, 1-2 digits. This value is converted to milliseconds and used by the service as the default timer value.

Available for both the manual and command line installation

#### IsGMT

Single digit 1/0 value indicating whether the local system should stay on UTC (GMT Casablanca/ Morroco Standard Time) time.

If enabled, the service does not attempt to check any XML file for scheduling information. Rather, it simply checks the time zone of the local system, and changes it to Morroco Standard Time (UTC) as necessary.

If disabled, the service will attempt to load the XML schedule file and determine the correct time zone for the local system from the file.

Not available on the command line (unattended) installation. The default value of 0 (false) is entered on the unattended installation.

#### VerboseLogging

Single digit 1/0 value. Enables or disables verbose logging for the service in the Windows Application log.

If disabled, the service will log events only during startup/shutdown, and if a time zone change was executed on the local system.

If enabled, the service will log all of the above, and in addition will log an event every time the timer lapses and a time zone check is performed.

Not available on the command line (unattended) installation. The default value of 1 (true) is entered on the unattended installation.

## Operations – Service Functionality

### Startup

On startup the system loads the Load registry settings into memory, determines the value of the timer, and starts the timer.

### On Timer Lapse

The service loads the xml file from the shared network location.

It then reads the global parameters section of the XML file. If necessary it updates any registry settings as specified.

It then reads the entry section to determine what time zone is currently in effect as per the schedule. It compares the local system time zone to the scheduled time zone, and if necessary updates the local system time zone to match the schedule.

## Operations - XML Schedule File

The XML file is divided into 2 sections as below. A sample XML file can be found in the Appendix of this document.

### Global Parameters section

This section performs two tasks as detailed below in the parameter definitions.

Second, it gives the administrator the opportunity to update the registry parameters on each client, affecting how the service is executed, without having to re-install the service on the client. If the administrator specifies any of the three values for XMLPath, FrequencyMinutes and VerboseLogging, the service will read those values and update the registry values on the local system, putting the new values into effect immediately. If nothing is specified in these values, they are ignored.

#### Schedule definition

The below parameter defines the type of schedule in the XML file, identifying it as either a **Weekly Recurring** or **Indefinite** schedule. See **Entry (schedule) items** section for schedule explanations.

###### IsRecurring

* A Boolean true/false value.
* If set to true, the service will process the entry items according to a 7-day **Weekly Recurring** schedule.
* If set to false, the service will process the entry items according to an **Indefinite** schedule.

#### Registry parameter update values

###### new\_XMLPath

A string value (without quotes) that indicates the new file path where the service is to look for the TimezoneSchedule.xml file. The service applies the change immediately, so it is necessary to have the new schedule XML file in the new location prior to updating this value.

###### new\_FrequencyMinutes

A single integer value, usually a number from 1-20, that specifies the timer value that the service should run under, in minutes. The service converts this number to milliseconds internally, so this number is entered by the administrator in minutes.

###### set\_VerboseLogging

A single digit 1 or 0 value, to enable or disable verbose logging.

### Entry (schedule) items section

##### Recurring Weekly Schedule

This would be useful on itineraries where the same 7-day voyage is run every week, and time zone changes occur on the same day/hour on a weekly basis, for example, where the ship goes hour forward / hour back each week in a circular voyage itinerary.

The system will read the below entries into a table, and sort them chronologically in order to determine which time zone is current for that particular point in the week.

###### day\_of\_week

A single digit numeric indicator that indicates what day of the week this schedule entry goes into effect on. See the following table for reference:

1 – Monday  
2 – Tuesday  
3 – Wednesday  
4 – Thursday  
5 – Friday  
6 – Saturday  
7 - Sunday

###### start\_time

The time of day that the time zone goes into effect, specified as a 24-hour military time in the format HH:MM. Therefore 6:00 PM would be specified as 18:00.

###### tzStandardName

The standard name of the time zone that goes into effect at the appointed time for this entry. Refer to the Windows registry for the Standard Name of the time zone that the system is changing to.

\*\* Note that the **Standard** name is **always** used when referring to the time zones. During daylight time, the local system will automatically adjust itself and set to the daylight version of the requested time zone if necessary.

For Windows time zones refer to the Windows registry at   
HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Time Zones

##### Indefinite Schedule

This is suitable on world cruises where a ship does not do repeating itineraries and rather would need to change to a particular time zone one-time during a cruise, and another time zone next cruise as the ship traverses over a long distance.

The service reads all entries into a table in memory and sorts them chronologically by date, in order to determine which time zone was most recently scheduled to go into effect.

###### start\_datetime

A date time string in the following format: YYYY-MM-DD HH:MM. Therefore the date of August 1, 2012 at 6:00PM would be specified as 2012-08-01 18:00.

###### tzStandardName

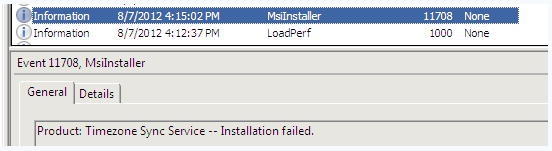
Same as above

## Installation

Installation can either be manually by double-clicking the TimezoneSyncSetup.msi package on the local system or via command line using command line parameters.

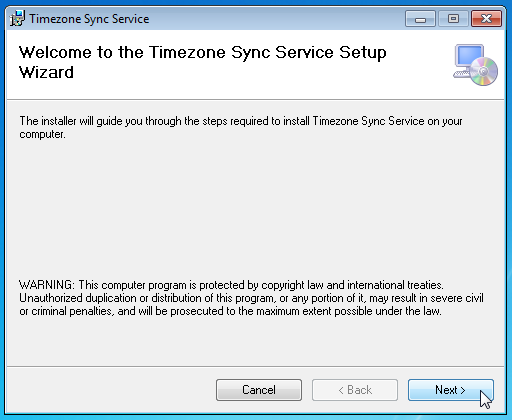
In either case the service will start automatically following the installation, so no restart of the local system is required.

\*\*\* Note that Microsoft DotNET Framework 3.5 is a prerequisite for installation. Windows will prompt during manual installation to download and install DotNET Framework 3.5 if necessary. During unattended installations, if the framework is missing the MsiInstaller will fail/exit logging Event ID 11708 (missing software dependency) and an event will be entered in the Application log similar to this example:

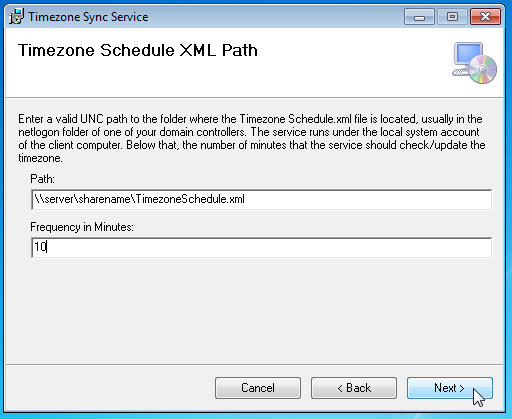


### Attended / manual installation

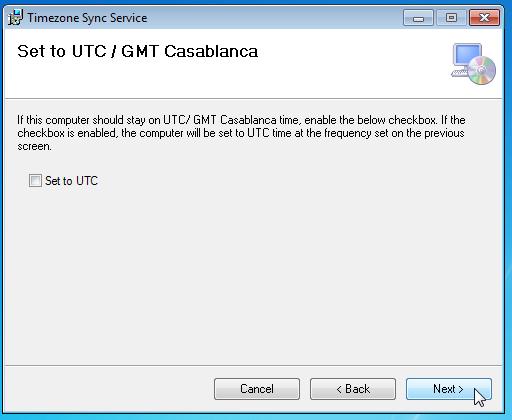
Initial launch screen of installation wizard. Click Next to continue.



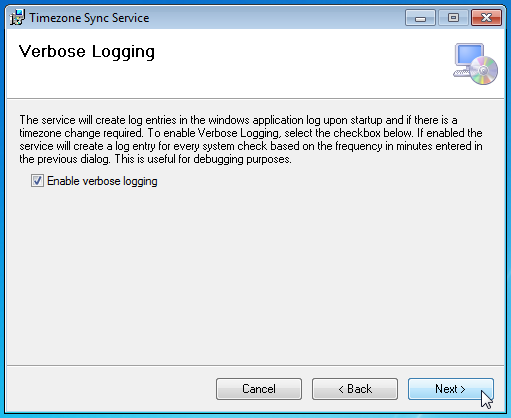
Enter the UNC path to the XML file containing the Time zone schedule entries. Secondly, enter the number of minutes that the service should operate for checking the time zone of the local system. The recommended default is 10 minutes.



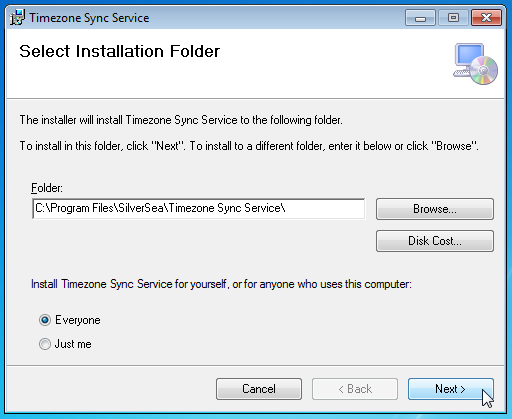
If the local system is meant to stay on UTC time, enable this setting here. Otherwise, leave empty and click next to continue.



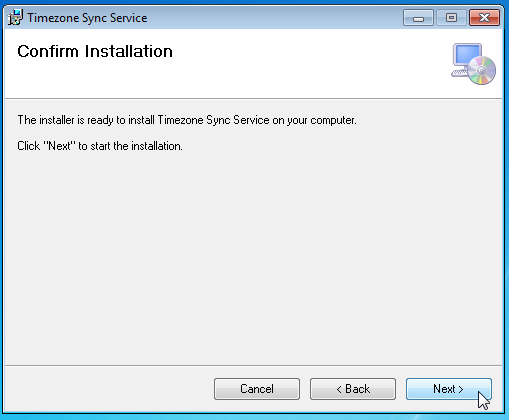
Verbose logging is enabled by default. It is recommended to leave this enabled until such time as the system is confirmed functioning, and then disable by updating the XML schedule file.

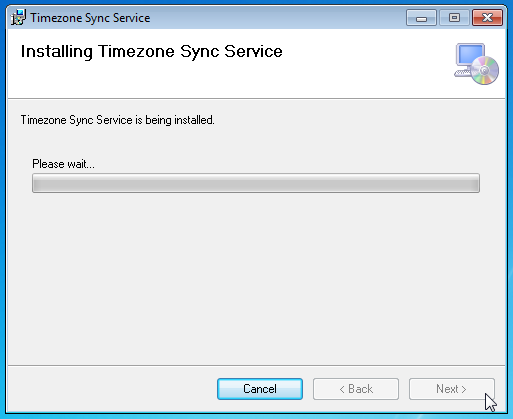


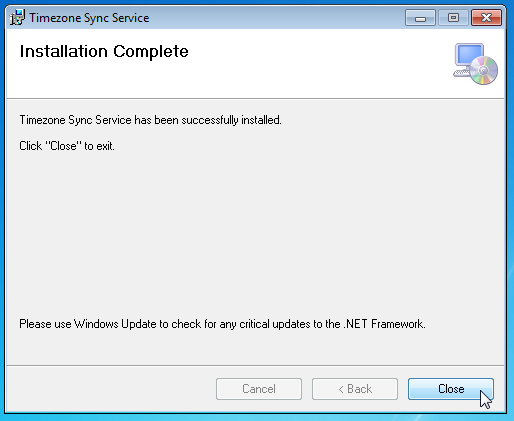
Confirm the default installation folder for the local system. Unattended installs will default to this folder.



Finally, confirm the installation and the install will complete.







### Unattended (command line) Installation

To perform an unattended install either from the command line or via a GPO or SCCM server, you need only specify the parameters for XMLPATH and FREQUENCYMINUTES. The IsGMT and VerboseLogging options are set to their default values (IsGMT = false, VerboseLogging=true).

##### Sample command line if using msiexec

msiexec /i TimezoneSyncSetup.msi XMLPATH=”\\unc\path\to\TimezoneSchedule.xml” FREQUENCYMINUTES=”10” /quiet

##### Sample command line without using msiexec

TimezoneSyncSetup.msi XMLPATH=”\\unc\path\to\TimezoneSchedule.xml” FREQUENCYMINUTES=”10” /quiet

### Uninstallation

Manual uninstallation is from add/remove programs (Windows XP/2003) or Programs and Features (Windows 7/2008) control panel options.

To uninstall by command line, you must use msiexec, as in the following example:

msiexec /x TimezoneSyncSetup.msi /quiet

# Appendices

## Appendix A: Sample TimezoneSchedule.xml

The below sample shows <entry> items setup on an Indefinite schedule. It is not necessary that the <entry> nodes be in chronological order.

### <?xml version="1.0" encoding="utf-8" ?>

<Schedule>

<global\_params>

<IsRecurring>false</IsRecurring>

<new\_XMLPath></new\_XMLPath>

<new\_FrequencyMinutes></new\_FrequencyMinutes>

<set\_VerboseLogging></set\_VerboseLogging>

</global\_params>

<entry>

<start\_datetime>2012-08-01 02:00</start\_datetime>

<tzStandardName>Pacific Standard Time</tzStandardName>

</entry>

<entry>

<start\_datetime>2012-08-05 02:00</start\_datetime>

<tzStandardName>Pacific Standard Time</tzStandardName>

</entry>

<entry>

<start\_datetime>2012-08-03 02:00</start\_datetime>

<tzStandardName>Alaskan Standard Time</tzStandardName>

</entry>

</Schedule>