The Windows™ Timestamp Project

G_HowTo_0260.pdf (V2.60)

The compressed archive *windowstimestamp_0260.zip* contains the following files:

- ./Samples

  Directory with a Microsoft Visual Studio 2013 solution G.sln.

- ./x86 (directory with x86 executables and libraries)
  - G_Kernel.exe
  - G_GUI.exe
  - G_IO_Service.exe
  - G_Setup.exe
  - G_dll.dll
  - G_Lib.lib
  - G_CreateServerSetupScript.exe

- ./x64 (directory with x64 executables and libraries)
  - G_Kernel.exe
  - G_GUI.exe
  - G_IO_Service.exe
  - G_Setup.exe
  - G_dll.dll
  - G_Lib.lib
  - G_CreateServerSetupScript.exe

- G_HowTo_0260.pdf
- G_Test.bat
- NTP_Server_Setup_200.bat
- G_LicenseKey.glk (optional, not included in the compressed archive)

The optional license file G_LicenseKey.glk may be reside in the packages home directory as shown here or copies of the license file may be put into the binary directories x86 and x64 respectively. However, the license file is not contained in the compressed archive. This license file is available upon request. A temporary license is sent by mail when it was requested during the download.

The package provides full interoperability between x86 and x64 parts. x64 clients may work with the x86 version of G_Kernel.exe and vice versa. Even x64 AND x86 clients may operate simultaneously. Note: The x64 and x86 versions carry the same names.

1 The files

1.1. G_Kernel.exe, the kernel

The package is launched by starting the file G_Kernel.exe. The required privileges are only obtained when this is done with administrator rights. Therefore G_Kernel.exe has to be run as administrator. This can be done by selecting the file with the right mouse button and choosing "run as administrator". Visit the Microsoft documentation to learn about other ways to obtain administrator rights. G_Kernel.exe can start in various modes and with various startup parameters. The modes and parameters are accessible through arguments supplied to G_Kernel.exe at startup. This can either been done in a console window or when starting G_Kernel.exe from another program. The modes/parameters are the following:

i. **G_Kernel.exe gui**

   Starts the package with a graphical user interface
   This is the default mode (no arguments)

ii. **G_Kernel.exe console**

   Starts the package with the console as output stream

iii. **G_Kernel.exe gui console**

   Both output streams are served.

iv. **G_Kernel.exe silent**

   Neither GUI nor console output will show

v. **G_Kernel.exe help**

   Supported arguments will be shown

vi. **ntp[=*Hostname/Address*]** starts the monitoring of the network time. The optional *Hostname* (e.g. "time-a.nist.gov") or *Address* as an ASCII string in internet standard dotted-decimal format (e.g. "178.23.124.2" or "2607:fe70:0:16::b") can be supplied to the ntp keyword to select a specific network time provider. The NTP monitoring starts at a default period of 2500 ms.

vii. **period=*UpdatePeriod*** The period keyword allows to select the NTP *UpdatePeriod* within 250 ms to 300000 ms (300 sec.). The default period is 2500 ms.

viii. **autoadjust** starts the service with the autoadjustment enabled. Note: The autoadjust keyword must be accompanied by the ntp keyword.

ix. **csv** dumps a csv file containing the collected filetime transitions during the initial analysis when used with *dev* flag. csv files are stored in /csv directory where the package resides. This directory is created automatically if it does not exist. This keyword is also required, when the qfprintf(CSV_FILE,...) library function is used. The "*G_Setup.exe status"* and/or the "*GUI Show Status Summary"* will add the list of configured NTP servers to the csv file.

x. **nolog** disables the output to a log file.

xi. **force_jump[=ms]** forces a instantaneous adjustment of the system time to match the first NTP read after startup. This keyword won't do anything until NTP data are available. This can be obtained by either starting in NTP/Autoadjust mode or set to these modes during operation. However, the first occurrence of an NTP offset will force the single adjustment only once. The optional value of ms results in continuous instantaneous system time adjustment whenever the NTP offset is beyond *ms*. However, force_jump with a threshold is ignored while "autoadjust" mode is active.

xii. **max_ntp_restarts=n** allows to configure the maximum number of restarts of the NTP service. A restart of the NTP service typically occurs when an NTP server turns out to be unreliable. A value of *n=0* will configure an endless amount of restarts. This may be advisable for unreliable networks.

xiii. **slow** allows to start in "slow" mode. The time service will automatically adapt to changes of the systems timer resolution. Note: The G_Kernel.exe "slow" mode is only supported for Windows 8 / Windows Server 2012 and above.

xiv. **nocalibration** skips filetime transition calibrations during startup and during operation.

xv. **cluster** starts G_Kernel.exe in cluster mode. Version 2.60 is capable to monitor a cluster of NTP servers.

Examples:

- *"G_Kernel.exe gui ntp=pool.ntp.org period=250 autoadjust"*

  This example starts with a single GUI instance, updating the network time from one of the servers in the pool of pool.ntp.org at a rate of 4 updates per second with autoadjustment enabled.

- *"G_Kernel.exe silent nolog"*
- *"G_Kernel.exe csv gui console ntp=time.windows.com period=350"*

G_Kernel.exe can only be stopped regularly by G_GUI.exe, which has a "Stop Kernel" button and by means of the G_Setup *exit* keyword. Of course G_Kernel.exe can also be stopped from within the task manager. While G_Kernel.exe has to run with administrator rights, G_GUI.exe and G_Setup.exe may also run as "non-privileged" processes. As such they are not allowed to terminate G_Kernel.exe. The "Stop Kernel" button is disabled when G_GUI.exe runs without privileges and G_Setup.exe will return an error when the keyword "exit" is used without privileges.

1.2. G_GUI.exe, the Graphical User Interface

The graphical user interface is a viewport to the behaviour of the G_Kernel.exe process. It can be started directly whenever the kernel runs. It can be ended anytime and restarted any time. Any number of GUI processes can be run simultaneously; however running hundreds of them is possible but not meaningful. The GUI is the suggested way to terminate the package ("Stop Kernel" Button with administrator rights).

For more information about the GUI visit www.windowstimestamp.com/description and http://www.windowstimestamp.com/news. G_GUI.exe depends on G_dll.dll.

The GUI supports the startup keyword **"tab=n"** to select the desired tab at the startup. Tab selection:

tab=1:	All Output
tab=2:	Error Messages
tab=3:	Calibrated Performance Counter Frequency Offset
tab=4:	NTP Offset
tab=5:	NTP Server Cluster (Only applicable in *cluster* mode)

Examples are shown in G_Test.bat (see 1.10 below). G_GUI.exe may be run with or without administrator rights. Running it without administrator rights has only one limitation: The "Stop Kernel" button and the "File -> Exit Kernel" menu item are disabled.

1.3. G_IO_Service.exe, the I/O

All blocking Input/Output is handled by this process. The kernel runs at very high priority and is therefore inadequate to actually do all kind of I/O. Log files are created and maintained by the I/O process too.

1.4. G_Setup.exe

The G_Setup.exe utility may be run from a console at any time when G_Kernel.exe is running. Also G_Setup.exe may be run with or without administrator rights. However, the "exit" argument is only permitted with administrator rights. This little tool allows to asynchronously perform modifications to the package setup:

Arguments:

- *"help"*	this text
- *"query"*	queries some current NTP and timer information.
- *"status"*	queries a broad configuration overview.
- *"ntp=pool.ntp.org"*	sets the NTP server host and/or selects it for use.
- *"ntp=178.23.124.2"*	sets the NTP server ip and/or selects it for use.
- *"server_add[=name/ip]"*	adds a new NTP server to the internal list of selectable servers. If no name/ip is supplied, G_Kernel.exe will automatically find a local NTP server to add.
- *"purge"*	purges the internal list of NTP servers. Not responding servers are removed. (Cluster mode: Cluster members cannot be purged.)

- *"period=2300"* sets the NTP update period in ms (currently supported 250 ms to 300000 ms).

- "mode=off" sets the mode of operation (Supported modes: "off", "monitor", "autoadjust").

- *"exit"* ends G_Kernel.exe. An alternative to the "Exit" button of the GUI. (Only supported with administrator rights.)

- *"sync_now"* forces an instant synchronisation of system time and NTP time. This is only applicable with modes *NTP* and *Autoadjust* enabled and locked. Note: This is similar to the startup argument *force_jump*.
  The accuracy of this synchronisation is not as good as the accuracy achieved with *Autoadjust* because it is based on a single NTP capture.

- *"timer_resolution=100000"* sets the systems timer resolution to 10 ms. This keyword is only available when G_Kernel.exe is started in "slow" mode. The service is disabled during startup and requires Windows 8 / Windows Server 2012 or above.

- *"sleep=2000"* suspends the execution of G_Setup keywords for 2 seconds.

- *"suspend=1000"* suspends the execution of a batch script for 1000 ms.
  Note: This is only applicable within batch scripts, it is NOT using G_Kernel.exe services!

- *"force_jump=threshold"* allows to asynchronously set the force_jump threshold.

- *"wait_kernel_accurate"* waits until G_Kernel.exe has reached accurate state (finished initial calibration).

- *"log="string""* writes the string to G_Kernel.exe log, will appear in the GUIs "All Output" tab. Note: The log string has to be enclosed in quotes when it contains spaces.

Arguments in cluster mode only:

- *"cluster_add[=name/ip]"* adds a new NTP server to the list of servers and marks is as cluster member. If no name/ip is supplied, G_Kernel.exe will automatically find a local NTP server to add. If the server is already in the standard server list, it will be marked as cluster member.

- *"cluster_remove=name/ip"*
  removes *name/ip* from the cluster. However, the server will remain in the list of configured servers.

- *"cluster/filter=filter"* adds f*ilter* to the list of applied filters.

- *"cluster/filter=-filter"* removes *filter* from the list of applied filters.

  Currently supported *filters*: "reject_outliers", "reject_rms_peaks", "none"

- *"cluster/schedule=mode"*
  sets the NTP query scheduling.

  Currently supported *modes*: "bunched", "distributed"

Examples:

- *"G_Setup query"*
  queries the current NTP configuration.

- *"G_Setup ntp=time.windows.com period=2500 mode=monitor"*
  establishes NTP monitoring with "time.windows.com" at an update period of 2500 ms.

- *"G_Setup ntp=178.23.124.2 period=250 mode=autoadjust"*
  establishes NTP autoadjusting with "178.23.124.2" at an update period of 250 ms.

- *"G_Setup mode=off"*
  disables the NTP functionality.

- *"G_Setup period=500 sleep=15000 sync_now"*
  sets the NTP Period to 500 ms, waits 15 seconds, and synchronizes to NTP

Remarks:

Arguments may be supplied in any order in upper or lower case letters. Parameters are updated asynchronously and may not take effect immediately. Operational details shall be looked at in the GUI and/or the log file. Arguments are processed in the supplied order, multiple occurrences of arguments are allowed. A brief example of G_Setup usage is given in G_Test.bat (see 1.10 below).

1.5. G_dll.dll, the dynamic link library

Many of the functions the kernel provides are accessible through the libraries. The GUI uses some of the functions and accesses them through the dynamic link library. G_GUI.exe needs this file. The dynamic link library is a wrapper to some of functions of the static link library; the number of functions available in the DLL is therefore a subset of the number of functions within the static library. It shall be noted that DLL function names are constructed by the function name with a subsequent underscore. Example: The library function Time(...) can be dynamically loaded from the DLL with the name Time_(...). See G_DllUser.c in the samples directory for more details on how to use the dynamic link library.

1.6. G_Lib.lib, the static library

See G_User.c in the samples directory for details about how to use the static c++ library.

1.7. G_HowTo_0260.pdf

This document.

1.8. Optional G_LicenseKey.glk, the license file

The whole package runs without the optional license key for a limited amount of time at calibrated accuracy. After this time, the calibration will be stopped. However, the package still runs without calibration, thus at reduced accuracy. Only G timer routines provided in the libraries are affected by an expired license. The entire NTP adjustment remains fully functional, it is independent of the license key. Possible license limits are shown in row two of the output stream or row three of the log-file (*ftime*). Optional NTP cluster services (in *cluster* mode) will stall upon expired license runtime.

1.9. The *Samples* Directory

Sample code is provided in various subdirectories here. The code is well commented and shall be self explanatory. Compilations made from this sample code can be run with the G package. A Visual Studio 2015 solution G.sln is located in the samples directory. This project includes sample code for G_User and G_DllUser. It also contains the library files for the dynamic link library G_dll.dll and the static library G_Lib.lib. Hint: Microsoft Visual Studio Community 2015 can be downloaded from Microsoft at no charge.

It is recommended to look at G_User.c and G_DllUser.c respectively to learn about how to use the libraries.

1.10. G_Test.bat

Batch file to give some examples on how G_Kernel.exe can be externally controlled. The batch file requires a parameter to branch into the desired executable directory. Example: "G_Test.bat x64". Note: This script has run with administrator rights.

1.11. G_CreateServerSetupScript.exe

This little tool creates an NTP setup script file as described in 1.11. One file created with this tool is NTP_Server_Setup_200.bat which is included in this package.

```
G_CreateServerSetupScript V2.60 (x64) usage:

G_CreateServerSetupScript.exe iterations delay [number [n x countrycode]] [max_stratum=n]

- iterations: number of pool scans, e.g. 10.
- delay: delay between poolscans in seconds, e.g. 20.
- number optionally specifies a limit for the number of servers to create.
- country code , e.g. "US". Multiple codes supported, e.g. "US DE SE FI".
  The country code may optionally be "local" to use the local country code only.
  "local" option is not allowed in combination with other country codes.
  Current local country code: "DE" (Germany).
- max_stratum=1..16 limits the search to servers providing at least the specified max_stratum.
```

```
Examples:      "G_CreateServerSetupScript 10 60
               "G_CreateServerSetupScript 10 60 max_stratum=2"
               "G_CreateServerSetupScript 10 60 GB PL ES"
               "G_CreateServerSetupScript 20 30 12 DE US SE FI"
               "G_CreateServerSetupScript 20 30 local"
               "G_CreateServerSetupScript 20 30 12 max_stratum=2"
               "G_CreateServerSetupScript 20 30 12 PT BR NO max_stratum=2"

Note: "Ctrl C" terminates properly with the number of servers collected at the time of
termination.
```

A log file (.txt) of the server search is produced at the time of termination:

```
12:17:55: Total server(s): 120, server hit stats: (hit rank, hits, country code, af_family,
stratum, host name, ip)

       1.  445   AM    IPv4, Ref: GPS  "ntp.amnic.net" (195.43.74.123)
       2.  254   SA    IPv4, Ref: GPS  "time1.isu.net.sa" (212.26.18.41)
               .
               .
     118.    1   US    IPv4, Ref: GPS  "darkcity.cerias.purdue.edu" (128.10.254.7)
     119.    1   PL    IPv4, Ref: PPS  "tempus1.gum.gov.pl" (212.244.36.227)
     120.    1   PL    IPv4, Ref: ATOM "ntp1.tp.pl" (80.50.231.226)

12:17:55: Total local (DE) "Germany" server(s): 12

       1.   12   DE    IPv4, Ref: MRS  "i4DF67E84.pool.tripleplugandplay.com" (77.246.126.132)
       2.   10   DE    IPv4, Ref: GPS  "ntp0.rrze.uni-erlangen.de" (131.188.3.220)

12:17:55: Sanity check OK.
12:17:55: End.
```

1.12. NTP_ServerSetup_200.bat

Script files created with CreateServerSetupScript.exe are using G_Setup.exe to establish a list of NTP servers within G_Kernel.exe. Their name is assembled dynamically depending on their content. They typically look like the following excerpt:

```
@echo off
echo setting up NTP server...
:: G_Setup script to configure NTP servers:
:: Note: None of the subsequent G_Setup command lines exceeds 4095 characters.
::
G_Setup server_add=196.25.1.9 server_add=41.248.247.207 server_add=193.108.227.130 ...
::
:: Total servers included: 200
echo 200 servers configured.
```

The setup script basically consists of calls to G_Setup.exe with a large number of server_add=xxx arguments. The setup is performed with more than one call to G_setup.exe when the length of the command line exceeds 4095 characters.

2 Testing

To make a long story short: Testing the package is not dangerous, just start G_Kernel.exe as administrator and watch the results. Toggle through the tabs of the GUI or even start more instances of the GUI to look at various tabs at the same time. Check/uncheck the NTP/Autoadjust check boxes to see what's going on.

Running the package will show how accurate timestamps can be obtained. The GUI also provides a small timed event test. The accuracy shown in the GUI shall get down to a few microseconds or less after a minute.

Note: Some security software, e.g. virus scanners, may delay the start of the suite when it is called for the first time. This may cause parts of the windowstimestamp suite to complain and to terminate with an error window. However, those security packages will keep the "test when run for the first time" in mind and subsequent starts will be successful. To overcome this hiccup, it is advisable to perform the virus test ahead of the first start.

We would like to encourage you to provide us with test results by means of mailing the files created in the \csv and the \log directory to info@windowstimestamp.com. You may inspect those files (plain text files) in order to verify that no personal information is forwarded to us. You may also take specific parts out of the files if you desire.

...