SIEMENS

siemens.com/wincc-open-architecture

Whitepaper

Discussion of WinCC OA time behavior

About this document

This whitepaper provides you with answers to the following questions:

- What are major principles regarding time synchronization in WinCC OA.
- What parameters / configs are relevant for time synchronization.
- How does WinCC OA work when the time synchronization fails.
- How does WinCC OA work in case of changes in time / time zones.

Disclaimer

Security information

Siemens provides products and solutions with industrial security functions that support the secure operation of plants, solutions, machines, equipment and/or networks. They are important components in a holistic industrial security concept. With this in mind, Siemens` products and solutions undergo continuous development. Siemens recommends strongly that you regularly check for product updates.
For the secure operation of Siemens products and solutions, it is necessary to take suitable preventive action (e.g. cell protection concept) and integrate each component into a holistic, state-of-the-art industrial security concept. Third-party products that may be in use should also be considered. For more information about industrial security, visit http://www.siemens.com/industrialsecurity.

To stay informed about product updates as they occur, sign up for a product-specific newsletter. For more information, visit http://support.automation.siemens.com.

Contents

This whitepaper gives an overview on principles and behavior of WinCC OA in case of failures of time synchronization as well as changes in time / time zones.

This document also includes UseCases which describe topics related to time synchronization and the behavior of WinCC OA.

WinCC OA is designed to work in distributed and complex SCADA architectures. Some principal architectures which are supported by WinCC OA will be explained. For example the following architectures are considered: Connection via driver to field devices, Remote UI, distributed System and redundancy System.
This document does not consider the topics of remote CTRL-Managers or remote WinCC OA drivers.

Normally it should be guaranteed that the whole SCADA System uses the same time synchronization. This can be realized with a GPS clock and the usage of a NTP-Server. ETM recommends using two GPS clocks (redundant) with a NTP server for complex SCADA systems.

This document is valid for WinCC OA 3.12 on Microsoft Windows operating system (Windows Server 2008 and for Clients Windows 7 and Windows 8).

1. Introduction

1.1. General problems of time rollback related to operating systems

A fundamental principle in computer programs is that time jumps in the past or the future have to be avoided. There are several side effects resulting from wrong time, which lead to severe problems. E.g.: Version control mechanisms will fail, when time is set to the past and "new" document versions are treated as versions, which were generated in the past. Or time jumps are often an indicator for security attacks, because replay attacks become possible. The effects become even worse, when several computers in a network are involved which are not time synchronized.

A second fundamental principle is, that time has to increase monotonously. That means messages are treated in the order of arrival. It's absolutely forbidden, that a message moves ahead of another. Imagine an operator or a control algorithm sends the commands open valve and then close valve but the valve receives the commands in opposite order: close and then open.

So all operating systems and standard software products recommend or even expect that systems are time synchronized. Several mechanisms are provided to handle small time differences in a defined way and there are algorithms available which allow synchronizing systems after they had different times. These algorithms allow a smooth approach to the correct time by avoiding time jumps, which lead to the above mentioned problems.

It has to be mentioned that those time tolerance mechanisms or time correction mechanisms are always problematic, since there is a certain period where the system stores values with a more or less wrong time stamp since the "correct" or "absolute" time is not known until the system or the total network is synchronized again to a reliable time reference.

That's why a system design with reliable time source and time synchronization is very important to avoid problems, when trying to analyze historical data.

Some examples from Microsoft support pages **[1]**:
A review of time rollbacks has shown that computers can adopt time that can be days, months, years or even tens of years in the future or in the past. The following issues can occur when computers roll forward or roll backward in time:

- Passwords on computer accounts, on user accounts and on trust relationships can be prematurely updated.
- Quarantines can be identified by NTDS Replication Event 2042 in Active Directory directory service replication.
- The mismatch of passwords is authoritatively restored for computer accounts, for user accounts, or for trust relationships. The recovery from such mismatches may require manual password resets on all accounts and trusts that are affected.
- Wrong time stamps of files and directories
- Wrong time stamps in log-files and databases

Windows operating systems include the Time Service tool (W32Time service) that is used by the Kerberos authentication protocol.
Kerberos authentication will work if the time interval between the relevant computers is within the maximum enabled time skew. The default is 5 minutes.

1.2. Behavior of WinCC OA

This chapter describes the standard behavior of WinCC OA in dealing with value changes with future and past timestamps.

- Internally WinCC OA uses the UTC time format.
- To display a time stamp the UTC time is recalculated according to the actual time zone and then displayed.
- WinCC OA works with time stamps with a resolution in milliseconds.
- WinCC OA messages are sequentially ordered per message source that guarantees that no message overtakes another message which was created later than the message before.
- Internally time stamps are stored with ns resolution, so that values with the “same” ms time stamp are handled in the correct time sequence.

To handle messages from the past, WinCC OA has a special mechanism: For a value with a time stamp before the last received value change, WinCC OA uses the latest source time of this value and adds a short time interval. So the corrected time is latest source time + interval. This value will be marked by “time invalid” status bit.

To solve the problem of wrong time synchronization some config entries exist. There are two groups of relevant config entries. One group defines the behavior for future value time stamps (valueChangeTimeDiff, validTimeDiff) and the other group defines the behavior for values in the past (negativeSourceTimeDiff).
These entries and the relation between them will be explained in following chapters.

What means future and past in WinCC OA?

Future in WinCC OA means a time stamp, which is larger than the system time of the WinCC OA server.

Past in WinCC OA always means a time stamp, which is older than the last time stamp of a data point element.

Example: If there are two datapoint elements X and Y. If X got its last value at 12:00 and Y got its last value at 14:00, than new value changes for X and Y at 13:00 would mean X would get a normal new value in monotonous order, while Y would get a value from the past.

<u>General behavior in WinCC OA:</u>
Received values with source time in the future will be accepted by WinCC OA with their source time.
Received past values (older than already accepted value changes) will be accepted by WinCC OA, but the time stamp will be replaced by the last known source time stamp of the DPE.

1.2.1. Values in the future:

Figure 1 shows the behavior of WinCC OA on receiving values with newer source time than its own server time.

1. If the value source time is between "now" and "now+valueChangeTimeDiff", WinCC OA will accept the value with its source time (1) and it will be marked as valid.
2. If the value source time is between the "now+valueChangeTimeDiff" and "now+validTimeDiff", WinCC OA will accept the value with its source time (2) but the value will be marked as time invalid.
3. If the value source time is bigger than "now+validTimeDiff", WinCC OA will ignore this value change – the value change is therefore not processed.

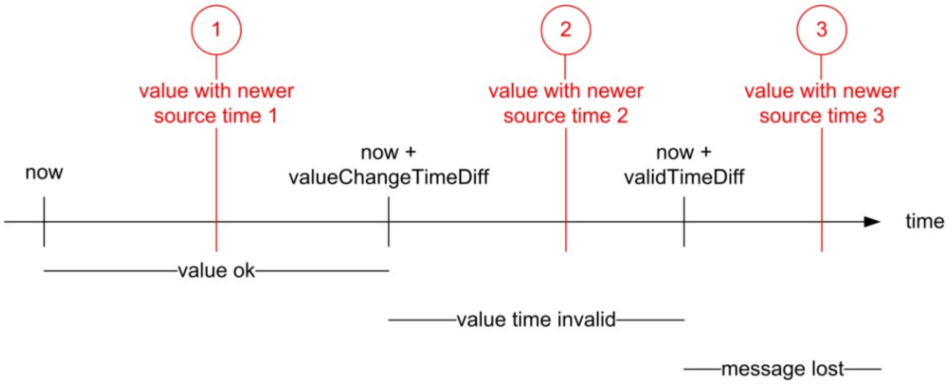

Figure 1: WinCC OA behavior for values in the future

1.2.2. Values in the past:

Figure 2 shows the behavior of WinCC OA on receiving values with older source time than the last source time for a DPE.

A DPE has a value with last source time (1).

1. If WinCC OA receives a value with source time between last source (1) and the range of negativeSourceTimeDiff (2), the value will be stored as valid with its last source time (1). For historical queries value changes with the same timestamps will be delivered in the right chronological order.
2. If WinCC OA receives a value with source time older than last source time-negativeSourceTimeDiff (3), the value will be saved with the last source time+1ms (4) and marked as time invalid.

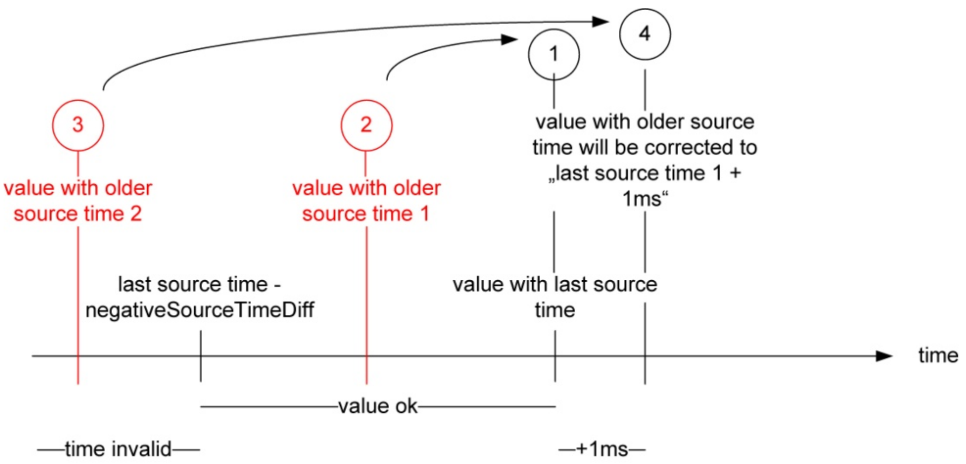

Figure 2: WinCC OA adopts the time stamp of values from the past

1.2.3. Conclusion of the value behavior in WinCC OA:

Figure 3 shows the overall behavior of WinCC OA and the status of the value and which time stamp will be saved.

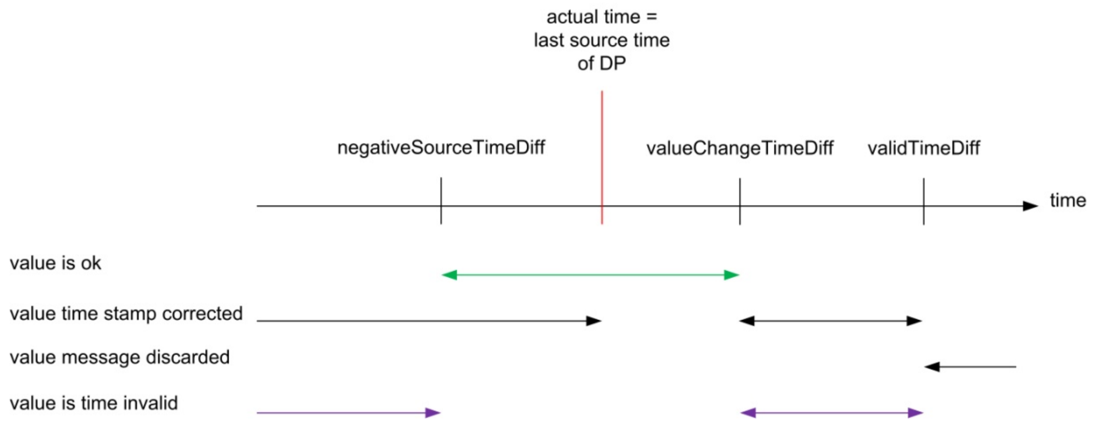

Figure 3: WinCC OA config entries and their relation to the status and time stamp of the value

Value is ok:

This means that the status bit of the value is ok.

Value time stamp corrected:

This means that the value will be saved with another time stamp as arrived in the WinCC OA system.

Value message discarded:

This means that the value message is discarded. The value change is not saved in the WinCC OA system.

Status set to time invalid:

This means that the status bit of the value is time invalid. One can identify those values in reports and trends.

1.2.4. WinCC OA redundancy behavior

This chapter describes how a redundant WinCC OA system processes values and how it behaves if time is not synchronized within the system.

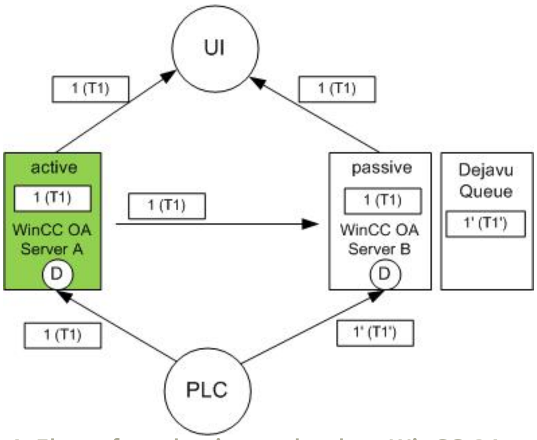

Figure 4: Flow of a value in a redundant WinCC OA system

Figure 4 above shows a redundant WinCC OA system, which receives values from a PLC with timestamp and provides these values to a connected User Interface. Furthermore the figure explains how value change messages are processed.

Following scenario applies:

1) The PLC sends a value (1)==(1') at 12:00 to both redundant servers
   a. The passive server B stores the received value (1') in its memory (in the so called "Dejavu Queue")
   b. The active server A forwards the value (1) from the PLC first to server B and then to the UI
   c. When the passive server receives the value (1) from the active sever, it updates its process image and forwards the value (1) to the UI which then only processes the value from the active server A.

2) The PLC sends a value (2)==(2') from the past (e.g.: time of PLC jumped two hours into the past)

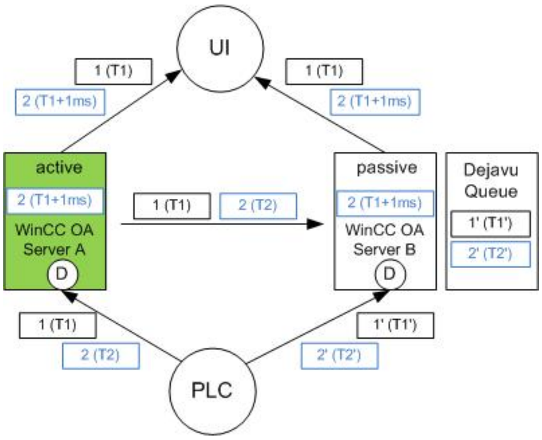

Figure 5: Processing a value from the past in a redundant WinCC OA system

   a. The passive server stores the original value (2') from the PLC into his "Dejavu Queue".
   b. The active server A

- forwards the original value (2) to the redundant partner B
- corrects the time stamp (current time stamp + 1ms),
- sets value to invalid and
- forwads the value with the corrected time and invalid status to the UI

c. When the passive server receives the value (2) from the active sever, it performs the same correction actions as server A (current time stamp + 1ms and invalid status) and forwards the value (2) to the UI

3) Redundancy Switch from active to passive server

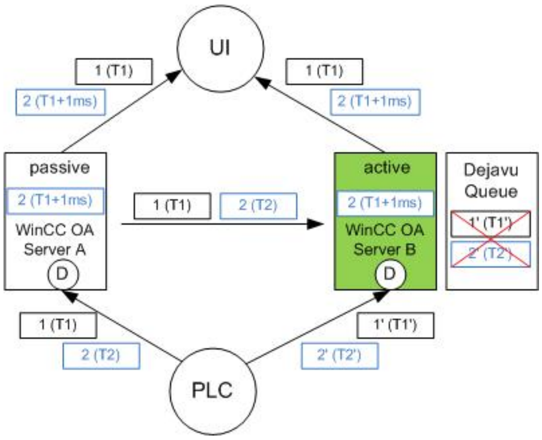

Figure 6: Performing a WinCC OA redundancy switch

If a redundancy switch occurs the new active server B compares the values of its Dejavu Queue to the current values of its process image. If the values in the Dejavu Queue are younger than the values in process image (e.g.: failure of former active redu host A, before it was able to forward the value to the former passive server B) the server B will update its process image by performing the same correction action as server A and notify the connected UI manager. All values in Dejavu Queue which are older than the current value in the process image are discarded.

4) The PLC jumped into the past and sends a value (4)==(4') after a redundancy switch occurred

When the new active server B receives a value (4') from the PLC, the Event manager compares the value (4') against the last received value in its process image and following rules apply:

a. If a General Query (GQ) is not performed (e.g.: deactivated via config entry) and the value (2) in the process image of server B is invalid, which is normally the case because per default WinCC OA markes values from the past as invalid, Server B accepts the next value change from the PLC (4'). The time stamps are corrected and values are marked as invalid until the time of the PLC and therefore also the time stamp of the next value (5') is newer than the time stamp of the last accepted value in the process image (4'). Table 1 shows a complete example from the viewpoint of Server B with real time stamps:

Sever Time	PLC Time	Todo	Time of accepted value	Accepted value	Value status	Value Origin
12:00	12:00	accept	12:00	(1)	valid	Server A
PLC jumps 2 hours into the past						
13:00	11:00	accept	12:00+1ms	(2)	invalid	Server A
Redu Switch from Server A to Server B						
13:45	11:45	accept	12:00+2ms	(4')	invalid	PLC
14:05	12:05	accept	12:05	(5')	valid	PLC

Table 1: Complete example, if current value is invalid from the viewpoint of Server B

b. A different behavior is intended, if the Genarl Query (GQ) is not perfomed and due to any reason, the last received value in the process image of Server B is valid (e.g.: due to config entry negativeSourceTimeDiff = 3h) and was received from its redundancy partner Server A. In this case the Event Manager discards all values from the PLC (e.g. values (4') and (5') in Table 2 below), because it has already received a valid and newer value (2) in its process image received from former active server A. Values are discarded until the time of the PLC and therefore also the time stamp of the next value (6') is newer than the timestamp of the last accepted value (2) in the process image of Server B. Table 2 shows a complete example from the viewpoint of Server B with real time stamps :

Time Server B	PLC Time	Todo	Time of accepted value	Accepted value	Value status at Server B	Value Origin
12:00	12:00	Accept	12:00	(1)	valid	Server A
PLC jumps 2 hours into the past						
13:00	11:00	Accept	12:00	(2)	valid	Server A
Redu Switch from Server A to Server B						
13:45	11:45	Discard		(4')		
13:50	11:50	Discard		(5')		
14:05	12:05	Accept	12:05	(6')	valid	PLC

Table 2: Complete example, if current value is valid from the viewpoint of Server B

c. If a Gernal Query (GQ) is performed by the driver after a redundancy switch (=WinCC OA default) the value (4') is marked with the General Query (GQ) flag. Those values are **always** accepted by server B, independed of their current status in the process image. Because of the time stamp of value (4') in the example below, the time stamp is corrected and the status is set to invalid. All following values from the PLC are accepted and processed. The time stamps are corrected and values are marked as invalid until the time of the PLC and therefore also the time stamp of the next value (5') is newer than the time stamp of the last accepted value (4'). Table 3 shows a complete example from the viewpoint of Server B with real time stamps:

Sever Time	PLC Time	Todo	Time of accepted value	Accepted value	Value status	Value Origin
12:00	12:00	accept	12:00	(1)	valid	Server A
PLC jumps 2 hours into the past						
13:00	11:00	accept	12:00+1ms	(2)	invalid	Server A
Redu Switch from Server A to Server B						
13:45	11:45	accept	12:00+2ms	(4')	GQ & invalid	PLC
14:05	12:05	accept	12:05+1ms	(5')	valid	PLC

Table 3: Complete example with General Query flag from the viewpoint of Server B

Conclusion:

With its default time behavior configuration (see chapter 2.5), a redundant WinCC OA system is able to process values with a timestamp from the past (or the future) even if a redundancy switch occurs. It is also not recommended to deactivate the GQ after a redundancy switch, because the GQ guarantees, that the current process value of the PLC is accepted by the WinCC OA system (as described in 2.4.4 point 4)

1.2.5. Alert behavior

This chapter describes the standard WinCC OA behavior regarding alerts, which are triggered by values with a time stamp in the past (see chapter 3.2, UseCase "Message delay from PLC").

In WinCC OA an alert is always created with the time stamp of the triggered value.

Example: If WinCC OA receives a value which is 1 hour in the past ($T_{current}$ - 1 hour), also the corresponding alert is created with the same time ("$T_{current}$ - 1 hour") – see Table 5.

This could lead to the following Example shown in Figure 7:

1. The last received value FALSE from the PLC was with PLC time stamp 11:00 at system time 12:00 (1).
2. At system time 13:00 the PLC sends a value TRUE with PLC time stamp 12:00 (2). This value generates an alert with ALERT CAME time 12:00.
3. The operator handles the alert and acknowledges (ALERT ACK) it 1 minute later at system time 13:01 (3).
4. The PLC sends the value FALSE at system time 13:30 with time stamp 12:30 which results in ALERT WENT with time stamp 12:30 (4).

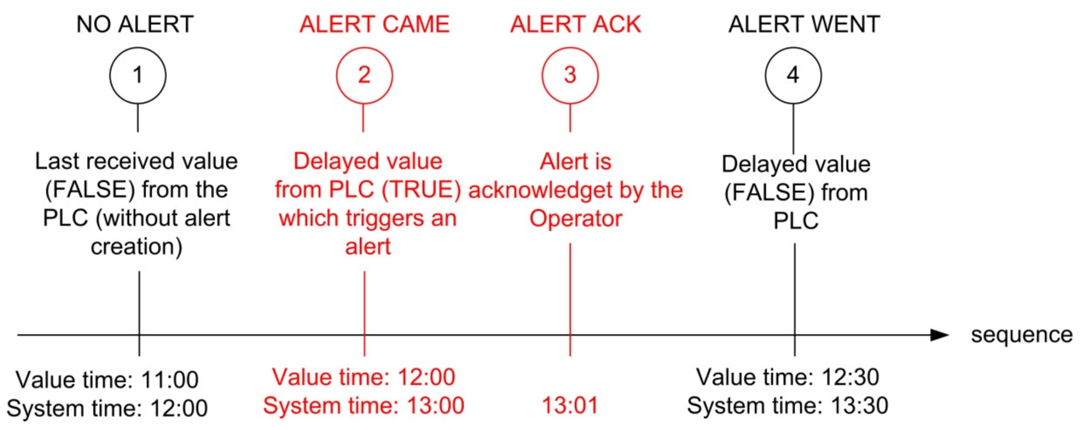

Figure 7: WinCC alert behavior

This would lead to following output (shown in table Table 4) in the WinCC OA alert screen. One could think that the operator needed more than 1 hour to acknowledge the alarm, where he needed only 1 minute in real time.

DP	PRIO	Direction	Alert Time	Acknowledge time	Acknowledge user
DP1	A	CAME	12:00	13:01	Operator1

Table 4: WinCC OA standard alert screen information with alert time stamp from the PLC

To overcome this szenario of "A blamed operator" WinCC OA additionally stores the current system time of the WinCC OA server for every alert, which means, one can find out, at which time the alert arrived in the WinCC OA system. The system time is stored in a special WinCC OA attribute and can be queried if necessary. So one can build following analyse screen to show the real progress of the alert

DP	PRIO	Direction	Alert Time	Alert System time	Acknowledge time	Acknowledge user
DP1	A	CAME	12:00	13:00	13:01	Operator1
DP1	A	WENT	12:30	13:30		

Table 5: WinCC OA additional alert system time for alert progress analysis

1.3. Config entries in WinCC OA

WinCC OA supports some config entries to solve the problem of time synchronization within their configured parameters. The relevance of each entry and their default settings will be explained in Table 6, which presents an extract of the WinCC OA Online Help [2].

The following config entries can be used to change allowed time differences for values in the future:				
Name	**Type**	**Default**	**Range**	**Comment**
validTimeDiff	uint (minutes)	10	0 - 2147483647	Maximum time in minutes, the source time of a value change may lie in the future. If the time lies further in the future, the corresponding ValueChangeMsg will be discarded and an error message is issued. Note Remote UIs are using an automatic adjusting so the validTimeDiff value will not be used, see Synchronization Client-Server. Note Should not be used for the correction of invalid time information of a PLC. For the purpose of compensating unreliable time information it may be advisable to use the driver time instead.

Name	Type	Default	Range	Comment
valueChangeTimeDiff	uint	30	>= 0	If the original value is changed, the Event Manager checks if the provided source time is more than valueChangeTimeDiff seconds in the future. In this case the system adjusts the source time to the current time and additionally sets the status bit invalid source time (_original.._stime_inv). Thereby, the system also sets the invalid bit. Redundancy: When a manager starts, it determines the time difference between its own computer and the computer of the server. If the system times differ for more than "valueChangeTimeDiff/2" seconds, the manager shows an error message. If the system times differ for more than "valueChangeTimeDiff" seconds, the manager closes the connection and shows an error message. If the valueChangeTimeDiff entry is set in the [event] section in older projects, the system shows an error message. The permitted time difference is 30 seconds. Caution: If a difference between the system times arises during the operation, the system does not check the difference. Caution: When changing valueChangeTimeDiff it is necessary to check the validTimeDiff value. valueChangeTimeDiff must be less or equal to validTimeDiff! Note Remote UIs are using an automatic adjusting independent of the valueChangeTimeDiff value, see Synchronization Client-Server.

The following config entry can be used to change allowed time differences for values in the past:				
Name	**Type**	**Default**	**Range**	**Comment**
negativeSourceTimeDiff	uint	0	>= 0	Time in seconds between one source time and the last source time without that the bit is set to "invalid source time". The source time will be corrected to "last source time + 1 nanosecond".

Table 6: WinCC OA config entries

Note:

These settings are for single WinCC OA Systems with Remote UIs or for standalone redundancy Systems with Remote UIs.

The default values of the config entries have been proven settings in many distributed and redundant projects. Significant changes have to be tested thoroughly.

To increase the tolerance for not synchronize PLCs one could set e.g.: negativeSourceTimeDiff = 30 (seconds).

Each WinCC OA driver must switch on “General Query” (GQ) in their config section.

The default value of validTimeDiff (10 minutes) is in accordance with recommendations for e.g. Active Directory or Kerberos (see Microsoft support pages).

1.4. Possible alternative behavior – Accept all time stamps

Description:

This chapter describes a theoretical alternative behavior and its consequences, which could be used for processing values which time stamps differ to the WinCC OA server time.

Instead of using the WinCC OA approach described in chapter 2.4.1 and 2.4.2, which ensures that values are written in their correct chronological order, the values could be accepted and written into WinCC OA with their delivered time stamp, even if this time stamp is in the past.

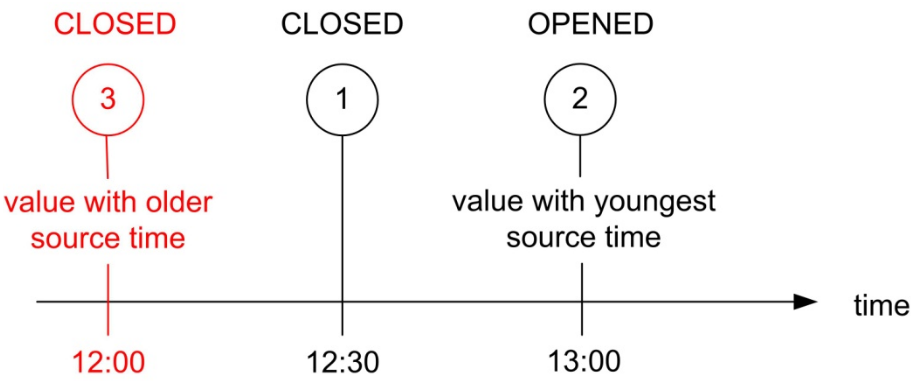

Figure 8: Alternative behavior: accept values with older time stamps

Example of a time jump at PLC shown in Figure 8:

1. At 12:30, a valve is "CLOSED" (1).
2. At 13:00 the valve is opened again and the status data point gets its last value with "OPENED" (2)
3. Afterwards the PLC time jumps into the past
4. The PLC sends the value "CLOSED" (3) at server time 14:00 with time stamp 12:00.

With this theoretical alternative approach, the value "CLOSED" (3) will be marked as time invalid and be written with time stamp 12:00 into the WinCC OA database.

Consequences:

1. For the given example, WinCC OA would not be able to decide which value should be shown to the operator in the User Interface. It is not clear, if one should see value "OPENED" or value "CLOSED" as actual value for the valve.
2. A historical query would provide a value trend, which differs from the real progress at the field (see Figure 9).

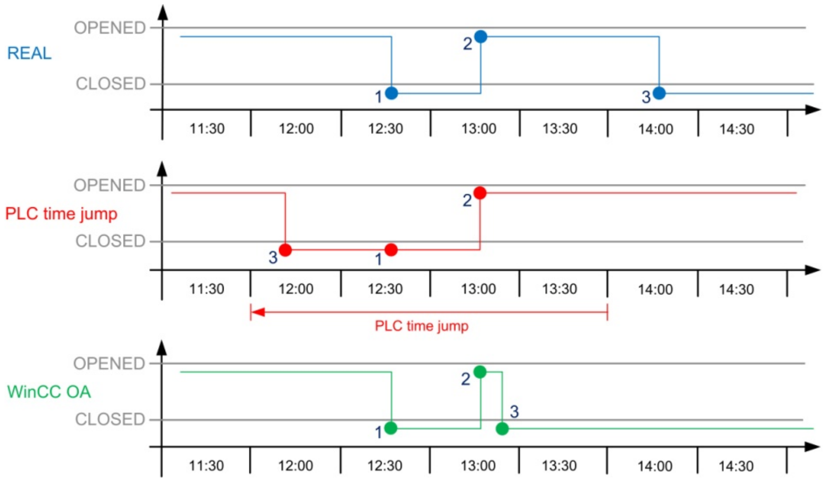

Figure 9: Trend with alternative behavior

Conclusion:
Due to noted consequences, the WinCC OA default behavior is the only useful approach to handle such values.

1.5. Behavior of operating systems / time synchronization software

This chapter describes how operating systems/time synchronization software solves the problem of time synchronization.

The standard behavior of the operating system or a special time synchronization software should be used.

The behavior is that operating systems as well as time synchronization software make no real jumps in the past if an external time synchronization signal is used.

This means that the clock only runs slower until it has reached the same time as provided by the external sources (e.g. GPS or over NTP).

Time changes in the future will be done as follows:

- Is the time change within the operating system parameters (see chapter 2.7.2), the system time will change in faster steps than in real life. This means that one minute on the PC is shorter than a real minute,
- Is the time change outside of the operating system parameters, the system time will change in one single time jump into the future.

Please note: Manual change of the system time is not a valid testcase to test the time synchronization mechanism (W32Time service or NTP). To test this behavior one would have to manipulate the external time reference.

1.5.1. NTP protocol

Another possibility to solve the problem of the time synchronization is to use the Network Time Protocol (NTP).

NTP is a networking protocol for clock synchronization between computer systems over packet switched, variable-latency data networks.

The protocol defines that every received time is a monotonic increase of the system time – time jumps in the past are not possible.

For more details – see RFC5905 - [4]

A timestamp T(t) represents either the UTC date or time offset from UTC at running time t. Which meaning is intended should be clear from the context. Let T(t) be the time offset, R(t) the frequency offset, and D(t) the aging rate (first derivative of R(t) with respect to t). Then, if T(t_0) is the UTC time offset determined at t = t_0, the UTC time offset at time t is

T(t) = T(t_0) + R(t_0)(t-t_0) + 1/2 * D(t_0)(t-t_0)^2 + e,

where e is a stochastic error term discussed later in this document. While the D(t) term is important when characterizing precision oscillators, it is ordinarily neglected for computer oscillators. In this document, all time values are in seconds (s) and all frequency values are in seconds-per-second (s/s). It is sometimes convenient to express frequency offsets in parts-per-million (ppm), where 1 ppm is equal to 10^(-6) s/s.

1.5.2. Important registry entries in the operating system

There are some parameters in the MS Windows registry which can be configured for the topic of time synchronization if you use the standard mechanism of the operating system (W32Time-service). The registry entries are located in:

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W32Time

To ETM´s knowledge these settings have not been changed in a productive system. As such ETM would recommend to take care when changing them and to test them thoroughly in a test system. The important entries will be listed here:

Registry Entry	MaxPosPhaseCorrection [1]
Value Type	DWORD
Subkey	HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W32Time\Config
Notes	This entry specifies the largest positive time correction in seconds that the service can make. If the service determines that a change larger than this is required, it logs an event instead. Special case: 0xFFFFFFFF means to always make the time correction. The default value for domain members is 0xFFFFFFFF. The default value for stand-alone clients and servers is 54,000 (15 hours).

Registry Entry	MaxNegPhaseCorrection [1]
Value Type	DWORD
Subkey	HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W32Time\Config
Notes	This entry specifies the largest negative time correction in seconds that the service can make. If the service determines that a change larger than this is required, it logs an event instead. Special case: -1 means always make the time correction. The default value for domain members is 0xFFFFFFFF. The default value for stand-alone clients and servers is 54,000 (15 hours).

Microsoft recommends that these entries should be reduced for Client settings. Dependent of time source, network and update interval as well security requirements. The recommended value is 3600 (=1hour) or smaller.

In Windows Server 2008, a default value for the MaxPosPhaseCorrection and MaxNegPhaseCorrection registry entries has been adopted. This default value is 48 hours [1].

Registry Entry	MaxPollInterval [1]
Value Type	DWORD
Subkey	HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W32Time\Config
Notes	This entry specifies the largest interval, in seconds, that is enabled for the system polling interval. Notice that, although a system must poll according to the scheduled interval, a provider can refuse to produce samples when samples are requested. The default value for domain members is 10. The default value for stand-alone clients and servers is 15.

Registry Entry	SpecialPollInterval [1]
Value Type	DWORD
Subkey	HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W32Time\TimeProviders\NtpClient
Notes	This entry specifies the special poll interval in seconds for manual peers. When the SpecialInterval 0x1 flag is enabled, W32Time uses this poll interval instead of a poll interval that the operating system determines. The default value on domain members is 3,600. The default value on stand-alone clients and servers is 604,800.

Registry Entry	MaxAllowedPhaseOffset [3]
Value Type	DWORD
Subkey	HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W32Time\Config
Notes	This entry specifies the maximum offset (in seconds) for which W32Time attempts to adjust the computer clock by using the clock rate. When the offset exceeds this rate, W32Time sets the computer clock directly. The default value for domain members is 300. The default value for stand-alone clients and servers is 1.

Registry Entry	PhaseCorrectRate [3]
Value Type	DWORD
Subkey	HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W32Time\Config
Notes	This entry controls the rate at which the phase error is corrected. Specifying a small value corrects the phase error quickly, but might cause the clock to become unstable. If the value is too large, it takes a longer time to correct the phase error. The default value on domain members is 1. The default value on stand-alone clients and servers is 7.

Registry Entry	UpdateInterval [3]
Value Type	DWORD
Subkey	HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W32Time\Config
Notes	This entry specifies the number of clock ticks between phase correction adjustments. The default value for domain controllers is 100. The default value for domain members is 30,000. The default value for stand-alone clients and servers is 360,000.

1.6. Behavior of Oracle RAC

Oracle RAC will be used in WinCC OA Systems as a high available data archive database. WinCC OA supports Oracle via RDB-Manager and Oracle Client.

An essential requirement from Oracle is that nodes of an Oracle RAC always must have the same time. This means that they will have one or more time synchronization sources but all nodes must have the same time.

A time stamp jump of one node is not allowed. [5]

1.7. Prerequisites

The following prerequisites have been set for a WinCC OA SCADA System:

PLC messages are sequentially correct, that means that no message overtakes another message which was created later than the message before.

Redundancy servers or servers with DIST connection must have the same time source which means that all WinCC OA servers have the same time.

Note: It is assumed that each device of the SCADA system can have time synchronization problems.

1.8. Limitations

If it is not possible to synchronize the whole SCADA system, you will face the following limitations:

- It is not allowed to use the dpSetTimed(), alertSetTimed() function in CTRL-Code. These functions use the time of the machine where the command is generated.
- Smoothing over time is not useable. Only smoothing over value works correctly.
- Hysteresis is not useable.
- Statistic functions are not useable.
- It is possible that during a time synchronization problem some limitations in the monitoring of the system including some differences in alert times to real time can appear.
- A manual change of the system time is generally not allowed.

This list does not claim to be complete; there may be some other restrictions which are not mentioned above.

2. UseCases

This chapter describes UseCases which explain the standard behavior of WinCC OA when the system time changes or will be adapted. The UseCases cover simple SCADA systems (PLC-Server-Remote UI) and more complex SCADA systems (e.g. distributed WinCC OA System).

2.1. Daylight saving and time zone changes

This chapter describes UseCases which explain the daylight saving time change and time zone changes.

2.1.1. UseCase: daylight saving time changes

Description:

The whole SCADA system makes the daylight saving time change each year.

Result:

The daylight saving time makes no problem in WinCC OA because internally the UTC time format is used.

Only for display purpose the UTC time is recalculated according to actual time zone and daylight saving settings. For displaying the following behaviors will appear:

Winter/summer time change: The 02:00 A.M. will not be shown in trends – only 01:00, 03:00 A.M.

Summer/winter time change: The 02:00 A.M. will appear twice in the trend.

2.1.2. UseCase: Different time zones

Description:

Each local control center has its own time zone.

Result:

Different time zones make no problem in WinCC OA because internally the UTC time format is used.

Only the displaying of the time information will be calculated according to the time zone which is configured in the respective operating system.

Only the UTC time will be saved. This means that no time zone information will be stored.

2.1.3. UseCase: Time zone change

Description:

The time zone of a local control center will be changed by an administrator via operating system mechanism.

Result:

Changes of time zones make no problem in WinCC OA because internally the UTC time format is used.

Only the displaying of the time information will be calculated according to the time zone which is configured in the operating system.

2.2. Time jumps into the past and into the future

General:

This chapter describes UseCases which consider time jumps into the past and into the future. Furthermore the UseCases describe manual and external (e.g.: GPS clock provides a wrong time) time changes.

Description:

The following table (Table 7) gives an overview, how WinCC OA works, if WinCC OA gets a wrong time signal, a manual time change is performed or PLC and WinCC OA system times are not synchronized.

UseCase	Time into past	Time into future
Single Server with wrong external time signal	OK[1)]	OK[2)]
Single Server with manual change of system time	OK[3)]	OK[4)]
PLC has wrong time	OK[4)]	OK[3)]
Behavior in REDU WinCC OA Systems	OK[4)]	OK[3)]
Behavior in DIST environments	OK[4)]	OK[3)]
Message delay from PLC	OK[4)]	OK[3)]

Table 7: UseCase Overview (definition of use cases see below)

Definition of Use Cases noted in Table 7:

Single Server with wrong external time signal:

A WinCC OA Server which is time synchronized via a GPS clock gets a wrong time signal from the GPS clock.

Single Server with manual change of system time:

On a WinCC OA Server a manual system time change is done. This case is not allowed in normal operation. A special security concept needs to be created so that a normal user cannot perform this action.

PLC has wrong time:

A PLC is not synchronized with the WinCC OA system time.

Behavior in REDU WinCC OA Systems:

A redundant WinCC OA System gets wrong external time signal. That means the two redundancy partners have different system times larger than a few seconds. It is a prerequisite that redundant WinCC OA System partners are time synchronized! A small time difference of a few seconds is tolerated.

Behavior in DIST environments:

One Server of a distributed WinCC OA System gets wrong external time signal. It is a prerequisite that all WinCC OA Systems are time synchronized (e.g. W32Time or NTP-Server) by the same time source! A small time difference of a few seconds is tolerated.

Message delay from PLC

A PLC messages will arrive in the WinCC OA System with some delay (e.g.: because of latency time, connection failure, values from PLC buffer, etc.).

[1)] OK, because the operating system supports monotonic increase of the system time, so jumps into the past are realized by making the system time run slower.

[2)] OK, because the operating system supports monotonic increase of the system time, so jumps into the future are realized by making the system time run faster.

[3)] OK, within standard WinCC OA parameters (valueChangeTimeDiff, validTimeDiff)

[4)] OK, within standard WinCC OA parameter (negativeSourceTimeDiff)

3. Conclusion

WinCC OA treats time changes in a similar way as other time sensitive software.

WinCC OA handles daylight saving time changes and time zone topics without any problems.

WinCC OA has built in tolerance mechanisms to handle not time synchronized data sources like PLCs.

WinCC OA marks value changes which had to be time corrected by setting a specific status bit so that these changes can be identified in the history.

It is strictly recommended to use a time synchronization software and provide a reliable time reference (GPS).

Disable manual system time manipulation for operators.

Ideally time synchronize PLCs if PLC is time stamping their values.

List of references

[1] Microsoft, "How to configure the Windows Time service against a large time offset," 11 09 2011. [Online]. Available: http://support.microsoft.com/kb/884776. [Accessed 12 01 2015].

[2] ETM professional Control GmbH, *WinCC OA V3.12 Online Help,* Eisenstadt, 2013.

[3] Microsoft, "Windows Time Service Tools and Settings," 05 2012. [Online]. Available: http://technet.microsoft.com/de-de/library/cc773263(WS.10).aspx. [Accessed 12 01 2015].

[4] Internet Engineering Task Force, "Network Time Protocol Version 4: Protocol and Algorithms Specification," June 2010. [Online]. Available: http://tools.ietf.org/html/rfc5905#page-8. [Accessed 12 01 2015].

[5] ORACLE, "Clusterware Administration and Deployment Guide," 2014. [Online]. Available: http://docs.oracle.com/cd/E11882_01/rac.112/e41959/admin.htm#CWADD838. [Accessed 12 01 2015].

ETM professional control GmbH - A Siemens Company
Marktstrasse 3
7000 Eisenstadt, Austria

www.siemens.com