Schneider Electric Software

Knowledge & Support Center

Doc Type	Tech Notes
Doc Id	TN52
Last Modified Date	02/20/2015

Time Zones and IndustrialSQL Server™ 8.0

LEGACY TECH NOTE #

378

SUMMARY

When submitting a Transact-SQL (T-SQL) query from a client application to IndustrialSQL Server, users typically request data using start- and end-times that reflect their local area time zone. The results should also be displayed in the expected time zone.

This tech note explains the characteristics of time zones in the context of IndustrialSQL Server 8.0 data storage and retrieval, and the behavior of such data when time zones are explicitly specified in a T-SQL query.

SITUATION

Operating System Time Zone Determination

InudstrialSQL Server 8.0 differs from earlier versions because all data stored is time stamped in Universal Time Coordinate (UTC) time, also known as Greenwich Mean Time (GMT). This is an absolute time and is not dependent on any Daylight Savings or regional settings in any way.

Generally speaking the information necessary to determine a local time is:

- Time Zone (Name and location)
- Is Daylight Savings to be applied?

A time zone contains two "offsets" relative to UTC: one offset that is used when Daylight Savings is in effect, and one when Daylight Savings is not in effect. For example, Pacific Time Zone contains two time zone offsets that called Pacific **Standard** Time (-8 hours offset), and Pacific **Daylight** Time (-7 hours offset).

In the Windows operating system, time zone information is stored in the registry at the following location:

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Time Zones]

This key defines each time zone by providing a string name and a time zone structure. This structure contains the information related to the two time zone offsets, and the information needed to calculate the time when the transition occurs.

The following time zone names are typical examples:

- Alaskan Standard Time
- Central Standard Time
- Egypt Standard Time
- Pacific Standard Time

Please be aware that time zone names may be different depending on the localized language of your operating system. However, the location in the registry for the names is the same for all languages.

Use SQL Server Query Analyzer to execute the following query against the Runtimedatabase. The query returns all available time zones on the local node:

SELECT * FROM TimeZone

	TimeZoneID	TimeZone	Description	Offset	RegistryName
1	1	Afghanistan Standard Time	(GMT+04:30) Kabul	270	Afghanistan Standard Time
2	2	Alaskan Standard Time	(GMT-09:00) Alaska	-540	Alaskan Standard Time
3	3	Alaskan Daylight Time	(GMT-09:00) Alaska	-480	Alaskan Daylight Time
4	4	Arab Standard Time	(GMT+03:00) Kuwait, Riyadh	180	Arab Standard Time
5	5	Arabian Standard Time	(GMT+04:00) Abu Dhabi, Muscat	240	Arabian Standard Time
6	6	Arabic Standard Time	(GMT+03:00) Baghdad	180	Arabic Standard Time
7	7	Arabic Daylight Time	(GMT+03:00) Baghdad	240	Arabic Daylight Time
8	8	Atlantic Standard Time	(GMT-04:00) Atlantic Time (...	-240	Atlantic Standard Time
9	9	Atlantic Daylight Time	(GMT-04:00) Atlantic Time (...	-180	Atlantic Daylight Time
10	10	AUS Central Standard Time	(GMT+09:30) Darwin	570	AUS Central Standard Time
11	11	AUS Eastern Standard Time	(GMT+10:00) Canberra, Melbo...	600	AUS Eastern Standard Time
12	12	AUS Eastern Daylight Time	(GMT+10:00) Canberra, Melbo...	660	AUS Eastern Daylight Time
13	13	Azores Standard Time	(GMT-01:00) Azores	-60	Azores Standard Time

Figure 1: TimeZones on the local node

Note that the offset values are included.

Fully Specifying a Local Time

It is important to understand the full specification of a local time and avoid the ambiguity during the period when time is "rolled back."

Note: "Rollback" occurs at different times of the year in the different hemispheres: in the Northern hemisphere the rollback occurs during the Fall, but in the Southern hemisphere, the rollback occurs during the Spring.

UTC is an absolute and continuous sequential time line, but during Daylight Savings Time changeovers, the local time is not always sequential nor continuous.

When performing the Daylight Savings Time conversion, ambiguity must be resolved (i.e. which 1:30 a.m. time stamp is correct?); this can only happen if the local time is fully specified. When Daylight Savings is applied, there is no one-to-one mapping between absolute and local time; this is exemplified by either a gap or an overlap in the local time line during the two changeover periods. If Daylight Savings is not applied, local time has a constant offset with respect to UTC and is fully specified by the time only.

Consider the case of the Pacific Time Zone in the Northern Hemisphere. The transition from Daylight Savings Time occurs at 2:00 a.m. on the last Sunday in October of every year. During the Fall transition there is an overlap of an hour. The time goes to **1:59:59** a.m. and then jumps back to **1:00:00** a.m. Consequently, the local time of 1:30 a.m. occurs twice on that day, an hour apart.

This time maps to two *different* times in UTC, and is the source of the ambiguity. The ambiguity is resolved by explicitly specifying the local time. Specifying the local time is accomplished by adding the time zone offset to the specified time and will result in the following 2 times:

- 1:30 a.m. **Pacific Daylight Time** [PDT] (for the first occurrence).
- 1:30 a.m. **Pacific Standard Time** [PST] (for the second occurrence).

These fully specified times are unambiguous and have a one-to-one mapping to UTC.

Specifying Time Zones for InSQL 8.0

To fully specify a local time, the time zone offset must be added.

In the above example, local time is not defined for the Spring changeover between 2:00 a.m. and 3:00 a.m. and has no corresponding UTC time. However, these times are defined if the local time is fully specified.

Some clarification is required here to avoid confusion with the above time zone definitions; we need to specify both the desired time zone and whether Daylight Savings needs to be accomodated. When submitting a query, a single string called **wwTimeZone** is used to specify both of these parameters within the context of the WHERE clause of the query.

The rule for creating the wwTimeZone string is as follows:

If the string contains the word "Standard" it will not adhere to Daylight Savings Time. If the word "Daylight" is present then Daylight Savings will be applied at the correct times where appropriate.

Using our example from the Pacific Time Zone, we may specify "Pacific Standard Time" or "Pacific Daylight Time". In this case the strings take on the following meanings:

- ***Pacific Standard Time***: Use the Pacific Time Zone and DO NOT apply Daylight Savings. This will always translate to a time zone offset of –8 hours (i.e. PST).
- ***Pacific Daylight Time***: Use the Pacific Time Zone and apply Daylight Savings at the applicable times. This will translate to an offset of –8 hours (i.e. PST) during the months from October to April, and an offset of –7 hours (i.e. PDT) from April to October.

Not all time zones observe Daylight Savings Time (i.e. "US Mountain Time" for Arizona). In these cases, specifying either "Standard" or "Daylight" will return identical results.

The number of time zones and their names may also be different depending on your operating system. For example, Windows 2000 makes a distinction between "Arab Standard Time" (Kuwait, Riyadh) and "Arabic Standard Time" (Baghdad), but Windows NT does not.

Query Specifications

Query Inputs

When specifying a query start and end time, a wwTimeZone is specified within the WHERE clause of the query. If wwTimeZone is *not* specified, a default time zone is used based on the operating system configuration settings on the IndstrialSQL Server node. In the case where wwTimeZone is specified,the time zone string is used as described in the section above.

When the retrieval engine of IndustrialSQL Server gets the start time and end time it is translated into UTC. Retrieval will resolve the ambiguities so as to give the widest possible time range. The start time will always default to the earliest UTC time and the end time to the latest UTC time. If a user requires to have more control of these times around changeover periods, UTC time can be used for the input times (the results will also be returned in UTC).

Query Output

Retrieval converts timestamps from disk into local timestamps as specified by the user in the query. The output time is not fully specified, so the user may need to use UTC to resolve ambiguities.

An obvious consequence of the conversion is that there will be data that overlaps during the Fall "roll back" changeover and a data hole for an hour during the Spring "roll forward" changeover. For example, a cyclic query that returns data every 20 minutes would get timestamps as returned below for the Fall changeover:

…
1:00 am Pacific Daylight Time
1:20 am PacificDaylight Time
1:40 am Pacific Daylight Time
1:00 am PacificStandard Time
1:20 am Pacific Standard Time
1:40 am PacificStandard Time
2:00 am Pacific Standard Time
…

Data returned for the Spring changeover:

…
1:00 am Pacific Standard Time
1:20 am PacificStandard Time
1:40 am Pacific Standard Time
3:00 am PacificDaylight Time
3:20 am Pacific Daylight Time
3:40 am PacificDaylight Time
…

Another consideration is when querying for data from a client workstation residing in a different time zone than the IndustrialSQL Server. A typical example might be someone in New York (Eastern Time Zone) wanting to query data from a server residing in California (Pacific Time Zone). If they wish for the data to be returned with respect to their physical location, they would need to specify their time zone in the WHERE clause as wwTimeZone='Eastern Daylight Time'. If they do not specify this, then the data will be returned with respect to the Pacific Time Zone (where the server is located).

C. Boutcher