

## Tech Note 741

# Solving Time Synchronization Issues for Historian IDAS and MDAS Tags

All Tech Notes, Tech Alerts and KBCD documents and software are provided "as is" without warranty of any kind. See the [Terms of Use](#) for more information.

Topic#: 002516  
Created: November 2010

## Introduction

Wonderware Historian supports a real-time window, which is defined as any value between -30 seconds and +999 milliseconds. This time window is the maximum delay relative to the current server time.

I/O servers such as DAServers, OPC servers, etc. supporting the SuiteLink protocol create a timestamp for plant data when the data is acquired.

It is important to synchronize the time between the I/O server(s), the computer clock for the IDAS(s), and the Historian.

## Application Versions

- Wonderware IndustrialSQL Server 9.0 and later
- Historian 10.0 and later
- Wonderware Application Server 3.1 and later
- Wonderware I/O Servers and DAServer
- Microsoft SQL Server 2005 and later

## Time Synchronization Best Practices

If you have multiple Historians on your network, synchronize the computer clocks for all of them to a single Historian that you designate as a time master server. You could then synchronize this master Historian to an external time source. Designate the master Historian using the **TimeSyncMaster** system parameter.

The Historian automatically and periodically synchronizes the computer clock of any remote IDAS with its own computer clock. The IDAS synchronization is enabled via the **TimeSyncIODrivers** system parameter.

Every hour, an IDAS will automatically synchronize the time stamping mechanism of any associated I/O Servers with its own computer clock. *It does not adjust the I/O Server computer clock.*

When a topic is disconnected/reconnected due to topic timeout or other communications failure, the I/O Server Time stamping is not updated until the time synchronization interval has passed. You can change this frequency of the synchronization using the **SuiteLinkTimeSyncInterval** system parameter.

**Note:** Time synchronization does not apply for data sources that use DDE protocol, since these data sources do not perform time stamping. Use the Historian time value for data coming via DDE protocol.

Historized Application Server Attributes (also called AppServer tags) stores data in the Historian via MDAS. For an MDAS-enabled client application, use the **NET TIME** command (**NET TIME\\computername /SET**) to synchronize the clients computer's clock with that of the time master Historian. This can be scheduled to repeat daily.

To schedule this, you can create a batch file and use the Windows Scheduler to run the batch file containing the NET TIME command periodically.

## Typical Message

When there is an issue relating to the time synchronization, you will see warning messages like the following:

## Time Sync Warning Message Type 1

**Attempt to store values in the future; timestamps were overwritten with current time (Briann5, 37232, 2010/11/08 12:21:12.108, 2) [SF38EDAP03; pipeserver.cpp; 2073; 318530]**

Where:

- **2010/11/08 12:21:12.108** is the current time on the Historian which was used to overwrite the timestamp by the source.
- **2** is the time, in seconds, that the supplied timestamp was ahead of the server time

## Time Sync Warning Message Type 2

**Values in the past did not fit within the realtime window; discarding data (Briann5, 37232, 2010/11/08 12:21:12.108, 2010/11/08 12:27:13.714) [CLVSRVR04; pipeserver.cpp; 2629; 3]**

Where:

- **Briann5** is the node sending data.
- **37232** is the **wwTagKey** value for the tag with the old timestamp.
- **2010/11/08 12:21:12.108** is the timestamp of the value received by the Historian.
- **2010/11/08 12:27:13.714** is the current time on the Historian.

## Time Sync Warning Message Type Summary

- When you see the **Attempt to store values in the future; timestamps were overwritten with current time** message, the Historian still stores data values but it overwrites the data time stamps with its own. You don't lose data in this case.
- When you see the **Values in the past did not fit within the realtime window; discarding data** message, the Historian is ignoring data because it is 30 seconds late or more. In this case, you are losing data and you need to correct the problem. However, you can use the same solutions listed below for both issues.

## Finding the Problem IDAS tag

Use the following MS query below. Replace the wwTagKey with the one you see in the message. In this example it is **37232** (following the previous example).

```
SELECT Tag.Tagname, IOS.ComputerName, IOS.ApplicationName, Topic.Name Topic
FROM Tag
INNER JOIN Topic ON Tag.TopicKey = Topic.TopicKey
INNER JOIN IOServer IOS ON Tag.IOServerKey = IOS.IOServerKey
WHERE Tag.wwTagKey = 37232
```

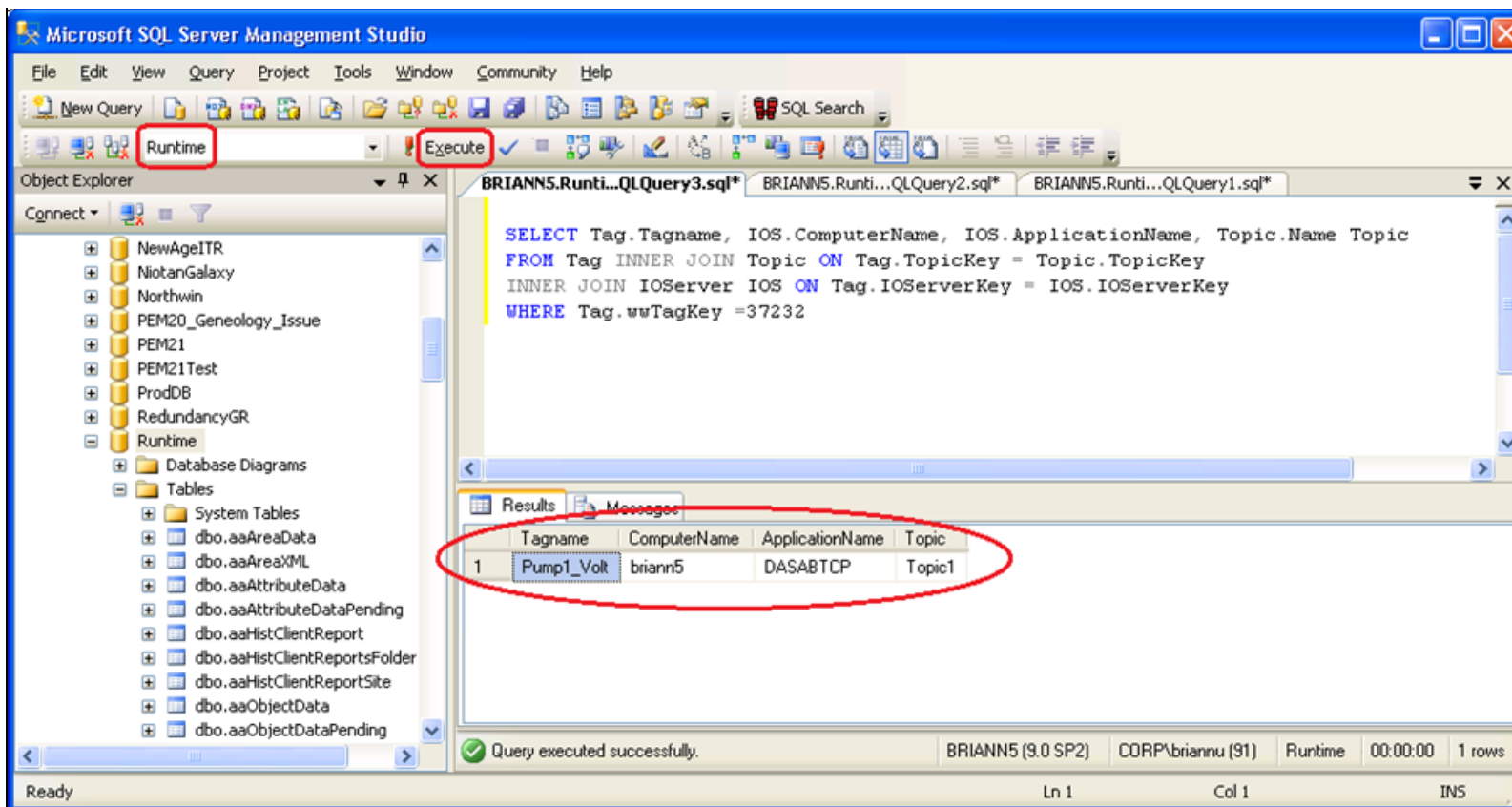


FIGURE 1: IDENTIFY THE TAG USING THE wwTAGKEY VALUE

To resolve this problem, you need to synchronize the system clock of the computer running the data source to that of the Historian.

## Finding the Problem MDAS App Server Tag

You can get the hierarchical name using the following SQL query:

```
SELECT t.wwTagKey, o.ContainedName, a.AttributeName AS TagName,
CASE WHEN p.aaTagName IS NULL
THEN a.AttributeName
ELSE p.aaTagName + '.' + o.ContainedName + substring(a.AttributeName, len(o.aaTagName)+1, 1000)
END AS HierarchicalName, o.aaTagName + '[' + o.ContainedName + ']' AS GroupName
FROM aaAttributeData a
LEFT JOIN aaObjectData o ON o.ObjectKey = a.ObjectKey
LEFT JOIN aaObjectData p ON o.ParentKey = p.ObjectKey
AND p.Type <> 0
LEFT JOIN Tag t ON t.TagName = a.AttributeName
WHERE t.wwTagKey=37225
```

The screenshot shows Microsoft SQL Server Management Studio with a query window open. The query is as follows:

```

SELECT t.wwTagKey, o.ContainedName, a.AttributeName AS TagName,
CASE WHEN p.aaTagName IS NULL THEN a.AttributeName
ELSE p.aaTagName + '.' + o.ContainedName + substring(a.AttributeName, len(o.aaTagName)+1, 1000)
END AS HierarchicalName, o.aaTagName+' ['+o.ContainedName+']' AS GroupName
FROM aaAttributeData a
LEFT JOIN aaObjectData o ON o.ObjectKey = a.ObjectKey
LEFT JOIN aaObjectData p ON o.ParentKey = p.ObjectKey
AND p.Type <> 0
LEFT JOIN Tag t ON t.TagName = a.AttributeName
WHERE t.wwTagKey=37225

```

The Results pane shows the following data:

	wwTagKey	ContainedName	TagName	HierarchicalName	GroupName
1	37225	testobject	testobject.IC_Ana2	testobject.IC_Ana2	testobject [testobject]

The status bar at the bottom indicates: Query executed successfully. BRIANN5 (9.0 SP2) CORP\briannu (71) Runtime 00:00:00 1 rows. The current cursor position is Ln 8, Col 54, Ch 54, INS.

FIGURE 2: DETERMINE HIERARCHICAL TAGNAME

To resolve the problem, you need to synchronize the system clock of the computer running the corresponding Application engine, and or the system clock of the computer hosting the data source to that of the Historian.

If you expect timestamps to come from the source (PLC, RTU, etc.) and that these will commonly be more than ~30-seconds old by the time they reach the Historian (not counting when the system is in "store forward"), use the **Late data** setting within the relevant Engines), which will enable the Historian to correctly store such values, but at the expense of lower throughput and more memory overhead on the Historian. You can minimize the impact on throughput if the data still fits within the **RealTimeWindow** system parameter setting on the Historian. The maximum setting is 5 minutes, but as long as most data still fits within this window, there should be minimal throughput impact.

B. Nguyen

*Tech Notes* are published occasionally by Wonderware Technical Support. Publisher: Invensys Systems, Inc., 26561 Rancho Parkway South, Lake Forest, CA 92630. There is also technical information on our software products at [Wonderware Technical Support](#).

For technical support questions, send an e-mail to [support@wonderware.com](mailto:support@wonderware.com).

[Back to top](#)

©2010 Invensys Systems, Inc. All rights reserved. No part of the material protected by this copyright may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying, recording, broadcasting, or by any information storage and retrieval system, without permission in writing from Invensys Systems, Inc. [Terms of Use](#).