WW INFO-01
Wonderware Historian 2012 R2

Elliott Middleton
Product Manager
Look back…
New Historian Unit Sales

Fiscal Year

2006 2007 2008 2009 2010 2011 2012

In System Platform
Standalone Historian
New Standalone Historian Revenue

First Full Year Of Historian 10.0

Growth Is Enterprise Licenses

- R-IDAS
- Enterprise
- Express
- Standard

Fiscal Year

- 2009
- 2010
- 2011
- 2012
New Large Tag Historian Units

First Full Year Of Historian 10.0

Fiscal Year

2009  2010  2011  2012

150,000
100,000
Historian Release Themes

**Historian 9.0**
- Powerful queries
- “Late” Data

**Historian 10.0**
- “Tiered” Historians
- Summary Tags
- Data Cleansing
Historian 9.0 Architecture

SQL Server

Retrieval

History Blocks

“Pull” Data Acquisition

Storage

“Push” Data Acquisition
Historian 10.0 Architecture

- **SQL Server**
- **Retrieval**
- **History Blocks**
- **Storage Engine**
- **Storage**
- **Replication Acquisition**

**Data Acquisition Processes**:
- "Pull" Data Acquisition
- "Push" Data Acquisition

**Key Components**:
- IDAS/SuiteLink
- Application Server
- Historian

**Additional Note**:
New "historian" hidden within 10.0
Replication

- Organized for faster retrieval
- More efficient use of disk I/O via caching
- Create tags & start collecting when Historian is offline
- Doesn’t “fragment” on updates
Storage Engine Retrieval Performance

Retrieval time for 3-months of 1-second data for 1 tag using “best fit” with 1-hour period*

*YMMV
The Most Amazing Wonderware Historian Ever

Continued

- World class desktop tools
- Rich query capability
- Low management cost & effort

All new integration with Application Server
Significantly higher tag counts

Redundant Historians

SQL Server 64-bit support (2008 R2 and 2012)

New Toolkit

Cyber security changes
Historian 2012 R2 Architecture

- **SQL Server**
- **Retrieval**
  - History Blocks
  - Storage Engine
    - "Pull" Data Acquisition
    - Classic Storage
    - "Push" Data Acquisition
  - Replication Acquisition
    - "Push" Data Acquisition

- Application Server <3.6
- Application Server >3.5
- IDAS/SuiteLink
- Historian

Most changes apply only here
SMC Status Panel Name Changes

**Historian 10.0**

<table>
<thead>
<tr>
<th>Module</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Storage</td>
<td>Started</td>
</tr>
<tr>
<td>Manual storage</td>
<td>Started</td>
</tr>
<tr>
<td>Storage2</td>
<td>Started</td>
</tr>
<tr>
<td>Replication</td>
<td>Started</td>
</tr>
<tr>
<td>Event system</td>
<td>Started</td>
</tr>
<tr>
<td>Retrieval</td>
<td>Started</td>
</tr>
<tr>
<td>Indexing</td>
<td>Started</td>
</tr>
<tr>
<td>OLE-DB provider</td>
<td>Started</td>
</tr>
<tr>
<td>Historian I/O server</td>
<td>Started</td>
</tr>
<tr>
<td>MDAS Server</td>
<td>Started</td>
</tr>
<tr>
<td>System driver</td>
<td>Started</td>
</tr>
<tr>
<td>Data acquisition on \emnsql10</td>
<td>Started</td>
</tr>
</tbody>
</table>

**Historian 2012 R2**

<table>
<thead>
<tr>
<th>Module</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Classic storage</td>
<td>Started</td>
</tr>
<tr>
<td>Classic manual storage</td>
<td>Started</td>
</tr>
<tr>
<td>Storage</td>
<td>Started</td>
</tr>
<tr>
<td>Replication</td>
<td>Started</td>
</tr>
<tr>
<td>Event system</td>
<td>Started</td>
</tr>
<tr>
<td>Retrieval</td>
<td>Started</td>
</tr>
<tr>
<td>Indexing</td>
<td>Started</td>
</tr>
<tr>
<td>OLE-DB provider</td>
<td>Started</td>
</tr>
<tr>
<td>Historian I/O server</td>
<td>Started</td>
</tr>
<tr>
<td>Client access point</td>
<td>Started</td>
</tr>
<tr>
<td>System driver</td>
<td>Started</td>
</tr>
<tr>
<td>Data acquisition on \MYHIST11A</td>
<td>Started</td>
</tr>
</tbody>
</table>
Upgrading

Upgrade Historian First
1. During upgrade, Engine goes into store-forward
2. After upgrade is complete, Engine forwards data and resumes
3. Engine continues using Classic Storage until it is upgraded
4. After Engine upgrade, using new Storage

No data loss in either scenario

Upgrade Platform First
1. After upgrade is complete, Engine immediately goes into store-forward
2. Remains in store-forward until Historian is upgraded
3. After Historian upgrade, using new Storage

Not Recommended
Application Server: Trivial To Historize

But...
Network/firewall challenges
Must be online to start
Historian tag data not always current
History loss on Engine failover
Engine Data Acquisition Throughput

Sustained: 50,000
Burst: 150,000
Late: 50,000

Values Per Second
What happens?

- Engine goes into store-forward
- New attributes not historized

- Engine goes into store-forward
- New attributes also enter store-forward
Historian-Application Server

What happens?

• All Engines exit store-forward at once
• Can overwhelm server & network

• Server manages store-forward exit
• Prevents overloading
Tag Meta Data
Tag Meta Data Updates
Scalability Targets

- Total number of tags in database: 655,360
- Number of licensed tags in database: 655,207
- License tag count: 0,000
- Total number of data values received: 57,324,025
- Overall data rate (per sec.): 153,359.31

Example test load on workstation hardware:

- CPU Usage: 4%
- Physical Memory: 26%
Reliable Data Acquisition

On failover, history gap from:
- Detecting failure
- Starting engine from checkpoint*
- Subscribing to I/O*
- Initializing history* N/A for 2012 R2

* Varies by number of objects

* Platform System
  2012

Redundant Engines
Redundant DI Objects

Historian
Client

Wonderware
Historian

Application
Server

Control System
High Availability

- Historian Client
- Wonderware Historian
- Application Server
- Control System

Multiple Clients
- Redundant Historians (2012 R2)
- Stratus
- VMware/HyperV Cluster

- Reliable Access
- Reliable Collection

- Redundant Engines
- Redundant DI Objects
Configuring Redundant Historians

MYHISTORIAN01

MYHISTORIAN02
Configuring Redundant Historians

[Diagram showing configuration settings for redundant historians]
Configuring Redundant Historians

- Server connection: MYHISTORIAN01
  - Server: MYHISTORIAN01
  - Authentication information
    - Login ID: wwUser
    - Password: ********
    - Remember password
  - Domain: 
  - Timeouts in seconds
    - Connection: 5
    - Query: 15

MYHISTORIAN01
MYHISTORIAN02
How It Works

A. Client retrieves “partner” name
B. On a failure, automatically switches

1. Engine retrieves “partner” name
2. Sends same data to “partner” with independent store-forward channels

Limitations
• No “self healing” of drive, history blocks, etc.
• Updates/inserts (SQL, CSV) must be repeated
• Client won’t switch on “store-forward”

H1 & H2 can be in the same or different locations
Engine Attributes

Bitmask: 0 0 0 0 0 0 0 0 0 1

Reserved  Partner  Primary

Logical “OR”
Historian Client Properties

Name of current server

Has a “partner”
Using Properties Script

InTouch

```plaintext
%tp = #aaHistClientTrend1.TagPicker;
%s = %tp.CurrentServer;
aahServer = %s.ServerName;
aahCurrent = %s.ConnectedServer;
aahHasPartner = %s.IsDualModeEnabled;
aahLoggedOn = %s.LoggedOn;
```

ArchestrA Symbol

```plaintext
```
How Does This Compare To:

High Availability Virtualization
- Con: Susceptible to software failures, OS reboots, etc.
- Con: Complex infrastructure, IT support required (SAN)
- Con: Special setup/software for geographically distributed solutions
- Neutral: License cost (more expense for geographically distributed)

Stratus FT Hardware
- Con: Susceptible to software failures, OS reboots, etc.
- Con: Premium hardware (cheaper to use conventional hardware x2)
- Con: Does not provide geographically distributed solution
- Pro: Single system to maintain
- Pro: No extra Historian license (redundant is +20%)
Historian 2012 R2 Communications

- SQL Server
- Retrieval
- Storage Engine
- Storage
- “Pull” Data Acquisition
- “Push” Data Acquisition
- COM/DCOM Named Pipes
- Replication Acquisition
- WCF Historian (Single TCP Port)
- SuiteLink (Single TCP Port)
- Application Server >3.5
- Application Server <3.6

Data Acquisition and Replication methods:
- “Pull” Acquisition
- “Push” Acquisition
AppServer & Historian Security

Can run under a different “Network Account”

Remote Galaxy user (e.g. “CORP\GalaxyUser”) 
• Must be able to authenticate 
• Must be in a local “aa_” group 
• No special rights needed

Network Account (e.g. “CORP\GalaxyUser”) must be a local administrator
“Data Compression” (aka “Filtering”)

Example
• Every fifth value is identical to the previous value
• 25% of remaining values are close to previous
• 50% of remaining values change at a steady rate
## Data Compression

<table>
<thead>
<tr>
<th>No Deadband</th>
<th>Value Deadband</th>
<th>Rate Readband</th>
<th>2012 R2</th>
<th>Rate Deadband</th>
</tr>
</thead>
<tbody>
<tr>
<td>80</td>
<td>60</td>
<td>30</td>
<td></td>
<td>30</td>
</tr>
<tr>
<td>80</td>
<td>60</td>
<td>60</td>
<td>60</td>
<td>30</td>
</tr>
<tr>
<td>80</td>
<td>80</td>
<td>80</td>
<td>80</td>
<td>30</td>
</tr>
<tr>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
</tbody>
</table>
Historian Editors In Application Server

Application Server 2012 (3.5)  Application Server 2012 R2 (3.6)

- Store forward deletion threshold:
- Store forward minimum duration:
- Forwarding chunk size:
- Forwarding delay:
- Buffer count:
- Enable Late Data:
  - Idle duration:
  - Process Interval:

- Enable compression:
- Throttling network bandwidth:
- Wait to send incomplete packets:
- Pre-processing buffer size:
- Store forward threshold:
- Store forward minimum duration:
- Enable Late Data:

Obsolete  Replaced  New  Unchanged
Historian Editors: Bandwidth Optimization

Packet = 64KB
~2,000 analog values

Saves ~20%
0 = unlimited
Bandwidth Limits

- **Unlimited (0)**
- **Physical**: 500
- **Streaming**: 200

Will thrash in and out of store-forward

- **Physical**: 500
- **Streaming**: 200

Available for store-forward

Available for other applications

Inadequate bandwidth
Engine Historian Editor

Advanced settings

Connection
- TCP port: 32568

Bandwidth optimization
- Enable compression
- Throttling network bandwidth: 0 kbps
- Wait to send incomplete packets: 1000 msec

Data management
- Pre-processing buffer size: 8 MB
- Store forward threshold: 100 MB
- Store forward minimum duration: 30 s
- Enable Late Data:
Storage Timestamp Limitations

**Historian 9.0**
- Late data
- Expanded Real-time (only for “late data” tags)

**InSQL 8.0**
- Real-time
- Future

Throughput

- **Historian 2012 R2 (AppServer)**
  - Time Series Data
  - “Now”

- “Values in the past did not fit within the real-time window”
- “Timestamp overwritten; values in future”

- Throughput: $\sim 10x$

Image: "The Industrial Software Revolution Begins Now!"
The “Late” Data Problem: Two Timelines

Example: constant 5-minute latency

Is it important to mark disconnects?

Complete solution is a “channel status” independent of the data stream

Affordable solution is injecting a disconnect indicator

Which timestamp to use?

Still requires data to be time-ordered
“Late” Data: One Timeline for Storage

Example: constant 5-minute latency

Source timestamp

No “gap” indication stored on disk

Store-Forward (3 minutes)

Stored On Disk

Server Time

8:00  8:05  8:10  8:15

Engine

Enable Late Data:
“Real Time” Data: Two Timelines

Example: constant 5-minute latency

Source timestamp

Stored On Disk

Server Time

Store-Forward (3 minutes)

Extended Store-Forward (2 minutes)

Which timestamp for “gap”? Out of sequence

Engine

Why is this a challenge?

Do not inject channel status
Comparison: “Real Time” vs. “Late”

Example: constant 5-minute latency

- Stored On Disk
  - Engine
  - No “gap” indication stored on disk
  - ☑️ Enable Late Data:

Extended Store-Forward (2 minutes)

- Stored On Disk
  - Engine
  - ☐ Enable Late Data:
Replication Server Changes

New

- Min SF Duration: 180 Seconds
- Buffer Count: 128
- Bandwidth: Unlimited

Works
Replication Naming Scheme

Format String:

\(<\text{HierarchicalName}>\) <TypeAbbreviation> <GroupAbbreviation>

Tag Name Example:


<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;TypeAbbreviation&gt;</td>
<td>TA</td>
</tr>
<tr>
<td>&lt;GroupAbbreviation&gt;</td>
<td>1H</td>
</tr>
<tr>
<td>(&lt;\text{HierarchicalName}&gt;)</td>
<td>Mixer_001.HoldingTank.InletPump.Motor.Switch.SerialNo</td>
</tr>
<tr>
<td>(&lt;\text{SourceTagName}&gt;)</td>
<td>MotorSwitch_001.SerialNo</td>
</tr>
</tbody>
</table>
Hierarchical Name Functions

TagName/Hierarchical Name Toggle

Query 100,000 tags
Using CSV Files For Data Acquisition

500 CSV files, each for 1,000 tags
64-bit Microsoft SQL Server Support

Historian Architecture

- OLE DB Provider runs in SQL Server process
- Some interprocess communication via shared memory
- 64-bit SQL Server → 64-bit Historian (mostly, but not all)

Solution in Historian 2012 R2:

- Native 64-bit Microsoft SQL Server Support
- Native 64-bit Historian OLE DB Provider
- 64-bit Process Address Space
- Ability to utilize all installed RAM

Higher scalability will require 64-bit

Phasing out support for case-sensitive collation
Total “On” Times For Each Of 3 Discrete Tags

SELECT DateTime, TagName, StateTime FROM History WHERE TagName LIKE 'Motor%' AND DateTime > '2012-07-04 9:00' AND DateTime <= '2012-07-04 9:30' AND wwRetrievalMode = 'ValueState' AND wwStateCalc = 'Total' AND wwCycleCount = 1 AND Value = 1

Historian 10.0
Historian 2012 R2
Total “On” Times For Each Of 3 Discrete Tags

```
SELECT StartDateTime, EndDateTime, TagName, StateTimeTotal
FROM StateSummaryHistory
WHERE TagName LIKE 'Motor%'
AND StartDateTime >= '2012-07-04 9:00'
AND EndDateTime <= '2012-07-04 9:30'
AND wwRetrievalMode='CYCLIC'
AND wwCycleCount=1
AND Value=1
```
Total “On” Time For All 3 Discrete Tags

SELECT DateTime
Total = SUM(StateTime)
FROM History
WHERE TagName LIKE 'Motor%' AND DateTime > '2012-07-04 9:00' AND DateTime <= '2012-07-04 9:30' AND wwRetrievalMode = 'ValueState' AND wwStateCalc = 'Total' AND wwCycleCount = 1 AND Value = 1
GROUP BY DateTime

✓ Historian 10.0
✓ Historian 2012 R2
CREATE PROCEDURE [dbo].[wwkbAllDiscretesOn]
    @startdate DATETIME2,
    @enddate DATETIME2,
    @tags NVARCHAR(MAX)
AS
    DECLARE @sql nvarchar(MAX)
    -- Generate a 'wide' query to total all the tag values, but subtract "1" from all but the last tag
    -- so that when all equal "1", the total will be "1"
    SET @sql = N'DECLARE DiscreteCursor CURSOR FAST_FORWARD
    FOR select DateTime, TagTotal from OPENQUERY(INSQL, ''SELECT DateTime, TagTotal=[' + REPLACE(@tags, ',', ',')]-1+['] FROM WideHistory
    WHERE DateTime >= '' + CONVERT(varchar(26), @startdate, 121) + ''
    AND DateTime <= '' + CONVERT(varchar(26), @enddate, 121) + ''
    AND wwRetrievalMode = "delta"')

    EXEC (@sql)

    DECLARE @RowTime DATETIME2
    DECLARE @RowTotal INT
    DECLARE @LastRowTime DATETIME2
    DECLARE @LastRowTotal INT
    DECLARE @TotalTime BIGINT

    SET @TotalTime = 0

    OPEN DiscreteCursor
    FETCH NEXT FROM DiscreteCursor
    INTO @RowTime, @RowTotal
    
    WHILE @@FETCH_STATUS = 0
    BEGIN
        IF @LastRowTotal = 1
        -- All the tags = 1
        SET @TotalTime = @TotalTime + DATEDIFF(MILLISECOND, @LastRowTime, @RowTime)
        SET @LastRowTime = @RowTime
        SET @LastRowTotal = @RowTotal
        
        FETCH NEXT FROM DiscreteCursor
        INTO @RowTime, @RowTotal
    
        IF @RowTotal = 1 AND @RowTime < @enddate
        SET @TotalTime = @TotalTime + DATEDIFF(MILLISECOND, @RowTime, @enddate)
    
    CLOSE DiscreteCursor
    DEALLOCATE DiscreteCursor

    SELECT StartDate = @startdate,
    EndDate = @enddate,
    Milliseconds = @TotalTime
GO
Time When All 3 Where “On” At Once

```
SELECT DateTime, AllOn=SUM([wwResolution])
FROM OPENQUERY(INSQL,'SELECT DateTime, [Motor1.State], [Motor2.State],
[Motor3.State], [wwResolution]
FROM WideHistory
WHERE DateTime > "2012-07-04 9:00"
AND DateTime <= "2012-07-04 9:30"
AND wwRetrievalMode="DELTA"
')
```
"wwResolution" For "Delta" Queries

### Historian 10.0

<table>
<thead>
<tr>
<th>DateTime</th>
<th>Motor1.State</th>
<th>Motor2.State</th>
<th>Motor3.State</th>
<th>wwResolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>2012-07-04 09:00:00.0000000000</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2012-07-04 09:00:10.1580000000</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2012-07-04 09:00:12.1570000000</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2012-07-04 09:00:15.1570000000</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2012-07-04 09:00:27.1580000000</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2012-07-04 09:00:31.2220000000</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

### Historian 2012 R2

<table>
<thead>
<tr>
<th>DateTime</th>
<th>Motor1.State</th>
<th>Motor2.State</th>
<th>Motor3.State</th>
<th>wwResolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>2012-07-04 09:00:00.0000000000</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>10158</td>
</tr>
<tr>
<td>2012-07-04 09:00:10.1580000000</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1999</td>
</tr>
<tr>
<td>2012-07-04 09:00:12.1570000000</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>3000</td>
</tr>
<tr>
<td>2012-07-04 09:00:15.1570000000</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>12001</td>
</tr>
<tr>
<td>2012-07-04 09:00:27.1580000000</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>4064</td>
</tr>
<tr>
<td>2012-07-04 09:00:31.2220000000</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>6938</td>
</tr>
</tbody>
</table>

Difference

The table above illustrates the "wwResolution" for "Delta" queries across two different historians, Historian 10.0 and Historian 2012 R2. The differences in the "wwResolution" values are highlighted, indicating the specific instances where the resolution differs between the two historians.
Exceeding A Threshold

Use with “ValueState” & “RoundTrip” retrieval
“ToDiscrete” Filter

```sql
select DateTime, vValue, StateTime, wwFilter
from History
where TagName in ('Reactor1Level')
and DateTime >= '2012-07-04 15:00:00'
and DateTime <= '2012-07-04 17:00:00'
and wwRetrievalMode = 'ValueState'
and wwStateCalc = 'Total'
and wwCycleCount = 1
and wwFilter = 'ToDiscrete(12,>)'```

✓ Historian 10.0
✓ Historian 2012 R2
Exceeding A Threshold For A Minimum Time

> 5 minutes

Level

“To Discrete”

Boundary
select DateTime, wwResolution
from History
where TagName in ('Reactor1Level')
and DateTime > '2012-07-04 15:00:00'
and DateTime <= '2012-07-05 17:00:00'
and wwRetrievalMode = 'delta'
and wwFilter = 'ToDiscrete(12,>)'
and wwResolution > 5*60*1000
Exceeding A Threshold For A Minimum Time

CREATE Procedure sp_GetAnalogToDiscreteData
    @DiscreteTag Varchar(33),
    @StartDate Time DateTime,
    @EndDate Time DateTime,
    @TargetTime Time DateTime,
    @Threshold Real,
    @CompareOperator Char(2)
AS

IF NOT EXISTS(SELECT TABLE_NAME FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_NAME = '#DiscreteEventTable')
Begin
Create Table #DiscreteEventTable(
    LeadingEdge Time DateTime NULL,
    TrailingEdge Time DateTime Null,
    TotalTime Real Null)
END
-- delete #DiscreteEventTable

DECLARE @LeadDateValue Time DateTime
DECLARE @TrailDateValue Time DateTime
DECLARE @DiffDateValue Time DateTime
DECLARE @DiffDateReal Real
DECLARE @DateValue Time DateTime
Declare @EventValue Real
Declare @EventDate Varchar (33)

-- Select data into a cursor record set

DECLARE Hist_Cursor CURSOR
FOR

IF @DateValue <= @TargetTime
    BEGIN
        SELECT #DiscreteEventTable Values (@LeadDateValue, @TrailDateValue, @DiffDateReal)
        END

IF @DateValue <= @TargetTime
    BEGIN
        SELECT #DiscreteEventTable Values (@LeadDateValue, @TrailDateValue, @DiffDateReal)
        END

END

Return all data to the calling user

Script courtesy of Ray Norman
Historian Toolkit Compatibility

Historian 9.0 Toolkit
- Continues to work with Historian 9.0-11.0
- No updates for this release
- Uses DCOM for communication & security

Historian 11.0 Toolkit
- Query history for all tags, including Summary Tags
- Only works with Historian 2012 R2 (aka 11.0)
- Only .NET 4.0, Application Server 2012 R2 (aka 3.6)
- Uses single TCP port for communications
- All new interface (though mapping is straightforward)
- Does not include tag listing (use SQL)
The Most Amazing Wonderware Historian Ever

All new integration with Application Server
Significantly higher tag counts
Redundant Historians
SQL Server 64-bit support (2008 R2 and 2012)
New Toolkit
Cyber security changes
Historian Client 10.1
Consistent Access To Strings Values
Unreachable Server
**Q:** What Is This?  

**A:** An “.aaTrend” File

```xml
<?xml version="1.0" encoding="utf-8"?>
<root>
  <tagList AAWRAPPED="5NHog3OiGJ1QEQtqWtreiKnuE+b0mZbhHmoH2UG3ZfjbtWbZetK74CZjODTXhyqugO91kNdF+E4ZshSK1MIVuLaxUZLEFZE1hvUYk7LMgc4j2C55SFyGGNhwNQv0pxWPxGAg/QW80GHLQJ8eX8UXZM1OTT+SnXfd14HiH73Ucf7bFGthKxV2mUt+zDIkNrpXYXyeLOeSsjxWLYRPj2nbq+UYqKZUAQTM8cu0JPCMF0z4rmEhjevmUTAC7EDWndZxeRUQZR5WjnR94Q3TioULXEmRiXa1GG8Rv17/cANJsnTS4pO+TlacgHANvFRnh4dFjwWPka0AzaKCqo71oJLE1q7WkYbkFiINHOevM48DmntsxIKUJt8Zv5AF0InJb89fWttQ3Z8BezFfbdUrv9fKep1/SlcAG6+taZ2dWcm2p/64W0Hs5/ipWoDp0f9oG0bP70hNGw/1fuBr9N5HuYdJ++
...
WYqRbG6Z/Plzi6V7nmGJzfasmwVjmUI1ZlB+8W3m3k5/yv9K75XaDxapZxmLVE5MNViMjGZLtrAwb0xG06GhpplmUkm51gB0P2aC9SLTPJNbw1nLwqTsIIVHK1qF7Gk9H0bQMU=" />
</tagList>
</root>
```
Historian Client 2012 R2 (10.1) Format

<?xml version="1.0" encoding="utf-8"?>
<root>
  <tagList VERSION="1" GRIDCOLUMNSTATE_VERSION="10SP1" DISPLAY_SINGLE_TAG="False"
    <retrievalOptions RETRIEVAL_STYLE_OPTION="BestFit-5" RETRIEVAL_STYLE_LAN
    <HierarchicalNameDisplay ENABLE="False"/>
    <threeD ENABLE="False" POINT_DEPTH="20" POINT_GAP_DEPTH="8" ROTATION="30"
  <cursorPen COLOR="#-65536" WIDTH="1" DASH_STYLE="Solid"/>
  <cursorPen2 COLOR="#6776961" WIDTH="11" DASH_STYLE="Solid"/>
  <timeAxis>
    <valueAxis VALUES="6" SHOW_GRID="True" GRID_LINES="2" SHOW_CURSOR="False"
    <cursorPen COLOR="#6774448" WIDTH="1" DASH_STYLE="Solid"/>
    <cursorPen2 COLOR="#6774448" WIDTH="1" DASH_STYLE="Solid"/>
  </valueAxis>
  <xValueAxis VALUE="30" SHOW_GRID="True" GRID_LINES="2" SHOW_CURSOR="False"
  <dataPointLabels TYPE="None" NUMBER="6"/>
  <traceGradient TYPE="OpacityGradient" STARTING_PERCENTAGE="20" ENDING_PE
  <background COLOR="#-1" GRADIENT_END_COLOR="#-1" IMAGE="" GRADIENT="None"

But, no documentation or support for direct XML changes

Some unsupported settings are exposed: Don’t change

Can still be opened by older versions
Thank You!

elliott.middleton@invensys.com