An Introduction to 
OPC UA
Open Platform Communication Unified Architecture

Dan Weise

rev 2b
What is the “window into the process”?

- HMI software or HMI panels are the “window into the process”
  - Operators supervise
  - Process data is displayed and recorded (historian)
  - Video monitors in control rooms and HMI panels everywhere

- It’s how one looks at
  - what is happening (HMI screen)
  - What has happened (historian)

- Integrators do the work of
  - Creating HMI screens
  - Populating HMI screens with specific data
    - Graphics, trend data, pushbuttons/switches, digital panel meters
  - Captive (in-house) or contract (independent)
What is involved in looking at the process?

- We are not integrators, so the tasks involved in getting an HMI package to display or log process data are not obvious.

What’s involved in making an HMI?

- Build a tag data base
- Get the devices to talk HMI software
- Configure an historian to log the data
- Develop screens and populate the screens

- The three in the red circle will involve the forthcoming OPC UA
HMI Tag Database

• Building the tag database
  – Spreadsheet of all the points, includes
    • Tag name
    • Data format (boolean, signed/unsigned integer, Real)
    • Engineering units
    • Scaling factor
    • Scale used in historian
    • Associated properties like alarm values, alarm status, PID values, auto/man
  – Can take days to weeks to build the tag database by hand
  – Some PLC/PAC devices export tag data to a spreadsheet .csv file
  – Spreadsheet has to be re-worked to be compatible with the HMI
# HMI Tag Database

Your Source for Process Control Instrumentation

## Tag Database

### Tags to show in the Editor

- Show All
- Invert

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Rows</th>
<th>Cols</th>
<th>Retentive</th>
<th>Init Value</th>
<th>Mod Start</th>
<th>Default</th>
<th>In Use</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>bnr750WReadTimeout</td>
<td>Boolean, 1D Array</td>
<td>1</td>
<td>10</td>
<td></td>
<td>0</td>
<td>400021</td>
<td>Decimal</td>
<td></td>
<td></td>
</tr>
<tr>
<td>bnrStatusRegisters</td>
<td>Integer, 16 Bit, 1D...</td>
<td>1</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td>Decimal</td>
<td></td>
<td></td>
</tr>
<tr>
<td>bnrHmi200WPositions</td>
<td>Integer, 16 Bit, 1D...</td>
<td>1</td>
<td>8</td>
<td></td>
<td>400021</td>
<td>400021</td>
<td>Decimal</td>
<td></td>
<td></td>
</tr>
<tr>
<td>System ID</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AR1516-000003(1)</td>
<td></td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>400021</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AR1516-000004(2)</td>
<td></td>
<td>1</td>
<td>2</td>
<td>500</td>
<td>100</td>
<td>400022</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AR1516-000005(3)</td>
<td></td>
<td>1</td>
<td>3</td>
<td>-100</td>
<td>1</td>
<td>400023</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AR1516-000006(4)</td>
<td></td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>200</td>
<td>400024</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AR1516-000007(5)</td>
<td></td>
<td>1</td>
<td>5</td>
<td>2</td>
<td>-300</td>
<td>400025</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AR1516-000008(6)</td>
<td></td>
<td>1</td>
<td>6</td>
<td>3</td>
<td>400026</td>
<td>400026</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AR1516-000009(7)</td>
<td></td>
<td>1</td>
<td>7</td>
<td>3</td>
<td>400027</td>
<td>400027</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AR1516-000010(8)</td>
<td></td>
<td>1</td>
<td>8</td>
<td>3</td>
<td>400028</td>
<td>400028</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>bnrHmi200WSpeeds</td>
<td>Integer, 16 Bit, 1D...</td>
<td>1</td>
<td>10</td>
<td>4</td>
<td>400029</td>
<td>400029</td>
<td>Decimal</td>
<td></td>
<td></td>
</tr>
<tr>
<td>bnr750WStatusRegisters</td>
<td>Integer, 16 Bit, 1D...</td>
<td>1</td>
<td>5</td>
<td>1</td>
<td>200</td>
<td>400024</td>
<td>Decimal</td>
<td></td>
<td></td>
</tr>
<tr>
<td>byarHomingModeSegments</td>
<td>Integer, 8 Bit Unsign. 255</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>400029</td>
<td>400029</td>
<td>Decimal</td>
<td></td>
<td></td>
</tr>
<tr>
<td>byarStepperResponse</td>
<td>Integer, 8 Bit Unsign. 255</td>
<td>1</td>
<td>1</td>
<td>10</td>
<td>400029</td>
<td>400029</td>
<td>Decimal</td>
<td></td>
<td></td>
</tr>
<tr>
<td>bcdHomingMode</td>
<td>Integer, 16 Bit BCD</td>
<td>1</td>
<td>10</td>
<td>4</td>
<td>400029</td>
<td>400029</td>
<td>Decimal</td>
<td></td>
<td></td>
</tr>
<tr>
<td>bWriteComplete</td>
<td>Boolean</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>400029</td>
<td>400029</td>
<td>Decimal</td>
<td></td>
<td></td>
</tr>
<tr>
<td>bWriteError</td>
<td>Boolean</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>400029</td>
<td>400029</td>
<td>Decimal</td>
<td></td>
<td></td>
</tr>
<tr>
<td>bWriteInProgress</td>
<td>Boolean</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>400029</td>
<td>400029</td>
<td>Decimal</td>
<td></td>
<td></td>
</tr>
<tr>
<td>bWriteSuccess</td>
<td>Boolean</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>400029</td>
<td>400029</td>
<td>Decimal</td>
<td></td>
<td></td>
</tr>
<tr>
<td>bWriteTimeout</td>
<td>Boolean</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>400029</td>
<td>400029</td>
<td>Decimal</td>
<td></td>
<td></td>
</tr>
<tr>
<td>bReadComplete</td>
<td>Boolean</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>400029</td>
<td>400029</td>
<td>Decimal</td>
<td></td>
<td></td>
</tr>
<tr>
<td>bReadError</td>
<td>Boolean</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>400029</td>
<td>400029</td>
<td>Decimal</td>
<td></td>
<td></td>
</tr>
<tr>
<td>bReadInProgress</td>
<td>Boolean</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>400029</td>
<td>400029</td>
<td>Decimal</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
HMI Tag Database

- What do you start with?
- Example: UDC 3200 Modbus

A.3 Loop Value Integer Register Map

The following table applies to the following instruments: UDC2300, UDC2500, UDC3200, UDC3300, UDC3500, DR4300 and DR4500. This table applies to Loops 1-24 except Loops 2-24 use the addresses shown in Table A-1.

<table>
<thead>
<tr>
<th>Address (hex)</th>
<th>Register (decimal)</th>
<th>Parameter Name</th>
<th>Access</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>40001</td>
<td>PV</td>
<td>R</td>
<td>Signed 16 bit integer  Prescale * 10 Note 5</td>
</tr>
<tr>
<td>0001</td>
<td>40002</td>
<td>RV: Remote Set Point; SP2</td>
<td>R</td>
<td>Signed 16 bit integer  Prescale * 10</td>
</tr>
<tr>
<td>0002</td>
<td>40003</td>
<td>Working Set Point</td>
<td>R/W</td>
<td>Signed 16 bit integer  Prescale * 10 Note 5</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>On a write the instrument will update the proper set point according to the loop’s currently selected set point.</td>
</tr>
<tr>
<td>0003</td>
<td>40004</td>
<td>Output</td>
<td>R/W</td>
<td>Signed 16 bit integer  Prescale * 10 Note 11</td>
</tr>
</tbody>
</table>

- Data in each cell has to be manually entered into each database cell;
  - no export. Very tedious. **Grunt work. Not a glamour task.**
HMI Communications

- Get devices to talk to the HMI software
  - Hardware links (the mules that carry the data)
    - Ethernet
    - RS-485
  - Protocols (the rules – who talks when)
    - Ethernet/IP
    - Profibus
    - Modbus
    - Proprietary
  - Software drivers do this task (like the Fedex logistics manager)
HMI Historian

Historian needs to know
- Which points to log
- How often to log data
- What number format it has to deal with
- Presenting a value – XXX, or XXX.X or XXX.XX or XXX

• Note
  - Tag list on the left
  - Properties of the tag in the red boxes
Tasks in HMI data display and logging

We’ve seen some of the effort involved in making an HMI tag and historian database

- Build a tag database
- Get the devices to talk to HMI software
- Configure an historian to log the data

- Develop screens and populate the screens – skip for this discussion

- The one in the red circle involves current OPC DA
- The one in the blue circle involves forthcoming OPC UA
What is OPC?

- OPC DA has been around since mid 1990’s (~20 years)
- OPC DA (Data Acquisition)
  - Very wide spread usage in process controls
- Primarily used by systems integrators
  - Runs in the background, not obvious
- Useable only in Windows products (not Linux)
  - based on DCOM, a Microsoft component
- OPC is all about data exchange
  - Read data
  - Write data
What is an OPC Server?

• OPC servers are ‘hidden’
  – they run in the background
• OPC server talks directly to digital ‘field devices’
• What kind of Field devices?
  – PLCs
  – PACs (HC-900)
  – HMI panels
  – Things that talk digital and need a ‘driver’, like a Modbus driver
    • UDA analyzer, UDC controller (talk Modbus)
    • Trendview
    • Siemens Coriolis FC410 (talks Modbus)
  – Does not include 4-20mA analog field instruments
    • Those need to connect to a system that talks ‘digital’
What is an OPC Client?

- OPC client is part of an HMI package
- HMI packages like
  - Wonderware
  - Iconics
  - Citect
  - RSView/FactoryView (Allen Bradley)
  - Win CC (Siemens)
  - Whatever GE calls Intellution nowadays
- When I hear ‘OPC Client’, I think HMI software
What is OPC DA?

• OPC DA is OPC servers talking to field devices and OPC clients
  – OPC server talks to field devices
  – OPC servers talk to OPC clients
  – OPC client talks to the HMI software
Rules to remember

• Rules
  – Any OPC server talks to any OPC client
  – OPC servers can talk to multiple devices and multiple OPC clients
  – OPC server needs a ‘driver’ for whatever field device it talks to
    • For instance, Kepware OPC server has an HC-900 driver that talks Modbus.
    • An OPC server can run multiple drivers
Why was OPC DA so successful?

- It made the task of getting data from field devices into an HMI much easier
- OPC server vendors specialize in drivers
  - Keep them updated
  - Their customer base is large enough to make writing drivers profitable
- OPC client vendors specialize in aspects of their HMI
  - do not have to bother with drivers
    - Division of labor pays off
- OPC clients talk to OPC servers
- Integrator is involved in configuring both OPC server and client
- Because OPC runs in the background, the OPC Foundation estimates that only 25% of end-users even know that OPC is involved in their HMI.
What is OPC UA?

- OPC UA is the Next Generation OPC
- UA is not dependent on Microsoft’s DCOM component
  - DCOM had its issues, configuration, security, robustness
  - Allows Linux based Honeywell One Wireless WDM gateway to use OPC UA
- UA exploits XML, a text based format for data exchange
  - Browsers, like FireFox, Chrome or Internet Explorer can read and display XML
- UA extends OPC to mobile devices, cell phones and tablets
- UA automates most of the effort in building a tag database
  - Recognizes and uses the properties of the tag, not just the value of the tag
OPC UA – tags with properties

- Remember all those things associated with a tag?
  - Tag name
  - Engineering units
  - Scaling factor
  - Scale used in historian
  - Report method – time based or report- o- exception
  - Associated elements like alarm values, alarm status, PID, auto/man
    - They’re all ‘properties’
What is OPC UA?

• For OPC DA, in some cases
  – all data/info had to be entered by hand
  – Many remaining properties had to be entered by hand
• OPC UA grabs all the properties along with the value
  – A tag has ‘properties’, not just a value
  – OPC server gathers the properties
  – OPC server transfers all the associated properties to the client
• Instead of manually entering any data for a database, the database will ‘auto-populate’ with all the tags and their associated properties
  – OPC UA: OPC server will pour the tag database into the OPC client

• This auto-populate function is well known within a vendor’s line
  – Migrating one from Plantscape to Experion PKS
• OPC UA promises auto-populate across vendor lines
Example 1 – what OPC UA might fix

- Display resolution is configured for an HMI
  - temperature values as whole numbers
  - Pressure values to 3 digits after the decimal point
Example 1 – what OPC UA might fix

- Same values displayed in a browser are raw floating point values

<table>
<thead>
<tr>
<th>Pen Name</th>
<th>Reading</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chg Rm DP</td>
<td>6.765264E-02</td>
<td>in H2O</td>
</tr>
<tr>
<td>Cmpd Rm DP</td>
<td>0.1295388</td>
<td>in H2O</td>
</tr>
<tr>
<td>Pass In 1 DP</td>
<td>8.973444E-02</td>
<td>in H2O</td>
</tr>
<tr>
<td>Pass In 2 DP</td>
<td>2.520309E-02</td>
<td>in H2O</td>
</tr>
<tr>
<td>Pass In 3 DP</td>
<td>5.660284E-02</td>
<td>in H2O</td>
</tr>
<tr>
<td>Pass In 4 DP</td>
<td>0.1096735</td>
<td>in H2O</td>
</tr>
<tr>
<td>Pass In 5 DP</td>
<td>8.999324E-02</td>
<td>in H2O</td>
</tr>
<tr>
<td>Pass In 6 DP</td>
<td>6.894965E-02</td>
<td>in H2O</td>
</tr>
<tr>
<td>Pass Out DP</td>
<td>6.913279E-02</td>
<td>in H2O</td>
</tr>
<tr>
<td>Comp Rm rH</td>
<td>34.87847</td>
<td>% RH</td>
</tr>
<tr>
<td>Staging DP</td>
<td>7.882208E-03</td>
<td>in H2O</td>
</tr>
<tr>
<td>Label &amp; Pack</td>
<td>63.10789</td>
<td>°F</td>
</tr>
<tr>
<td>Fin Prod Vault N</td>
<td>74.11005</td>
<td>°F</td>
</tr>
<tr>
<td>Fin Prod vault S</td>
<td>71.25351</td>
<td>°F</td>
</tr>
<tr>
<td>Lab Room</td>
<td>74.72208</td>
<td>°F</td>
</tr>
<tr>
<td>Lab Refrig</td>
<td>39.83991</td>
<td>°F</td>
</tr>
</tbody>
</table>

- On the recorder screen the value is 0.067, in IE the same value appears as 6.76526E-02
- On the recorder screen the value is 63, in IE the same value appears as 63.10789
  - Exponential format (how convenient)
  - Resolution to 5 or 6 digits after the decimal point (it’s mostly noise)
Example 1 – what OPC UA might fix

• Display resolution is a property, has already been defined:

• But IE does not access that property; IE accesses only the value and the engineering units
• Browser has no clue what the Trendview display resolution is
• OPC UA will grab all the object properties
• OPC UA will display the value in any venue the same manner (at the same display resolution) as defined in the source
Example 2 – what OPC UA might fix

- Selecting which tags or points appear on an HMI screen

- Although the pens have been named with tags (inlet pressure, inlet temp, outlet temp, demand, flow rate, total) selection is primitive, by pen #.  
  - The property ‘tag name’ isn’t available
- The easier it is to pick and choose correctly (because items are identified as people know them), the more the ‘database’ will be exploited
How will OPC UA manifest itself?

- Apps for mobiles will extend functionality
  - Definition of OPC is “data transfer to HMI’s”
  - HMI’s are portables:
    - Tablets
    - Cell phones
    - Widgets for your desktop computer
- Expanded use through its evolution
  - The movement away from a Microsoft requirement (no more DCOM)
  - Use of XML for a data language
  - mobile O/S can exploit OPC UA, Linux O/S (XYR6000)
- Reduce the cost of developing control room HMI’s
  - Automated build by auto-population of the tag database
- OPC UA is the backbone of IIoT
  - Industrial Internet of Things
**Why isn’t it here now?**

- Reason #1 software
- Reason #2 software
- Reason #3 software
- Latest version of Wonderware is OPC UA
- What version does Abbott run? (answer: 5 year old version, not OPC UA)
- We’re waiting the mobile and desktop apps which exploit OPC UA.
- Hold your breath, it’s coming.
Overview of OPC DA and OPC UA

Questions?