

Data Quality Management

Presented by:
Gordon MacLachlan (Mac)



Presentation Highlights

- ◆ Defining Data Quality and identifying issues
- ◆ FieldServer features important in analyzing data quality
- ◆ Data Arrays explained
- ◆ Role of Active mapping in data quality management
- ◆ Monitoring device status
- ◆ Data Typing
- ◆ Timing
- ◆ Read/Write Management
- ◆ Data Quality Management alternatives

Defining Data Quality



Data Quality is good if it is:

“fit for intended use in operations,
decision making and planning”
[Wikipedia]

Defining Data Quality

Attributes used to measure data quality:

- Accuracy
- Correctness
- Timeliness (Data Age)
- Completeness
- Relevance

FieldServer Data Quality Management Features

◆ Accuracy

- Data can be stored in relevant data format



FieldServer Data Quality Management Features

◆ Correctness

- Data can be viewed in FieldServer for evaluating correctness.
- Configurable mapping allows for easy data matching.

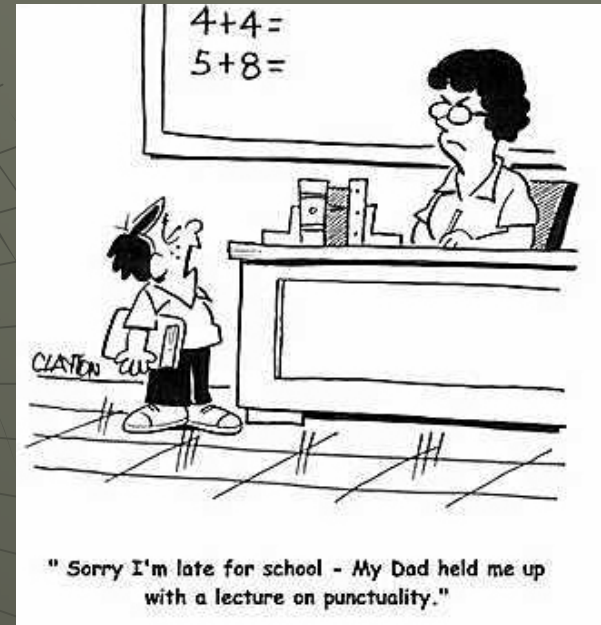


Correct!

FieldServer Data Quality Management Features

◆ Timeliness (Data Age)

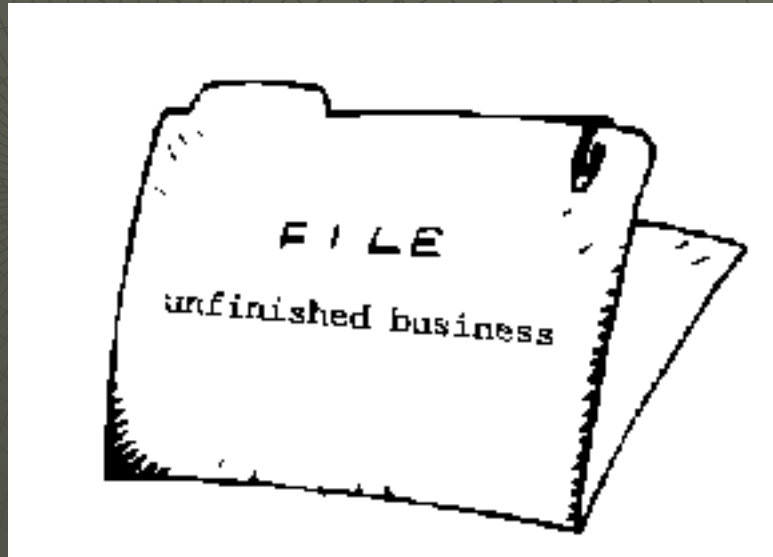
- Data Currency can be monitored and flagged for "out of limits"
- Timing can be manipulated.



FieldServer Data Quality Management Features

◆ **Completeness**

- Configuration parameters allow for the population of missing information where protocols allow it.



FieldServer Data Quality Management Features

◆ **Relevance**

- Configurable mapping allows for selective data transfer.



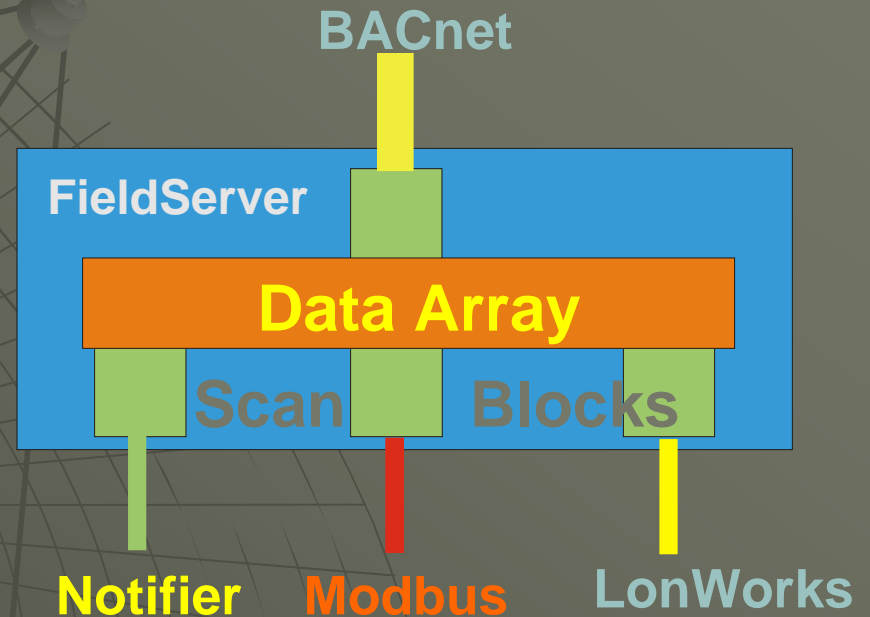
Factors that Influence Data Quality

- ◆ Data Control
- ◆ Data Age
- ◆ Data Types
- ◆ Device Availability
- ◆ Data Structure
- ◆ Read/Write Management
- ◆ Communications Timing

Data Arrays in Detail

What is a Data Array?

- ◆ Data Storage buffer for data transferred through the FieldServer
- ◆ Accessible to all FieldServer drivers for transfer of data, thereby providing the "link" between protocols



Data Arrays in Detail

What is a Data Array?

- ◆ Contains the following properties:
 - Name for unique identification (15 characters, configurable)
 - Length (Declares number of data items to be stored in Array)
 - Data Type (Declares format for storage)

```
//-----  
//  
//   Data Arrays  
//
```

```
Data_Arrays  
Data_Array_Name , Data_Format      , Data_Array_Length  
Status_DA      , Float            , 200  
Alarm_DA       , Bit              , 600
```

Data Arrays in Detail

What is a Data Array?

- ◆ Number of data arrays that can be declared is limited only by available memory (which is large)
- ◆ Number of data items per data array that can be declared is limited only by available memory (which is large)
- ◆ Point count is almost always exceeded before Data Array capacity is exceeded, so these limits are mostly theoretical.

Data Arrays in Detail

What can a Data Array do?

- ◆ The quality related to the communication of data stored in a data array can be monitored if the data is associated with an active mapping (active map descriptor)
- ◆ The age of data within a data array is monitored for last update.
- ◆ Data Age of a data array can be used to flag bad data if the age of the data exceeds a certain limit.

Data Arrays in Detail

What can a Data Array do?

- ◆ Data can be moved between arrays for the purposes of grouping data to improve relevance.
- ◆ Data can be moved to larger Data Arrays to provide larger buffers for mass communication.
- ◆ Data can be moved with special moves that create relevance or accuracy (e.g: floating point moves or Logic)

Data Arrays in Detail

**What does a Data Array look like?
Concept Diagram:**

Name: VMA_AI

Offset 0: 26

Offset 1: 28

....

Offset n: 1.2

Data Arrays in Detail

What does a Data Array look like?
RUI presentation:

```
Remote User Interface
Data Array
Data Array 1 / 1
Data_Array_Name CURR_PORT_DATA
Data_Array_Length 100

Data format: UInt16
Bytes Per Item: 2
Data age 31:09.653s
Oldest 31:09.662s

 0: 1 0 0 0 0 8843 0 0 0 0
10: 0 0 0 0 0 0 0 0 0 0
20: 0 0 0 0 0 0 0 0 0 0
30: 0 0 0 0 0 0 0 0 0 0
40: 0 0 0 0 0 0 0 0 0 0
50: 0 0 0 0 0 0 0 0 0 0
60: 0 0 0 0 0 0 0 0 0 0
70: 0 0 0 0 0 0 0 0 0 0
80: 0 0 0 0 0 0 0 0 0 0
90: 0 0 0 0 0 0 0 0 0 0

Keys: <+> Next/Prev <Ctrl+> Last/First <PgDnUp> More <G>oto <ESC>
<H>ex <D>ec <U>nsigned Fl<o>at B<y>te <B>inary <S>tring <M>odify <+> Offset
```

Active Mapping

What is an active mapping?

- ◆ FieldServer Terminology: Active Map Descriptor/
Responsible Map Descriptor
- ◆ An active mapping controls the communications interaction with the remote device.
- ◆ Active mappings are responsible for communications timing and communications monitoring.

Active Mapping

What is an active mapping?

- ◆ Look for rdbc, wrbc, wrbx in the function field to identify active mappings.
- ◆ Passive mappings only respond to communications requests/updates made by active mappings.

//=====							
// Temperature Mapping in Deg_F							
Map_Descriptors							
Map_Descriptor_Name	Data_Array_Name	Data_Array_Offset	Lon_Function	Function	Note_Name	SNVT_Type	SNVT_Units
nviSpaceTmp	DA_Temps	0	NVUI	Server	EC001	SNVT_temp_p	Deg_F
nvoSpaceTmp	DA_Temps	1	NVUOIMX	wrbx	EC001	SNVT_temp_p	Deg_F

Passive Mapping

Active Mapping

Active Mapping

What can an active mapping do?

- ◆ Control communication timing
- ◆ Monitor the health of a communications session

Timing control

//=====								
// NMFETCHC Mapping								
Map_Descriptors								
Map_Descriptor_Name	Data_Array_Name	Data_Array_Offset	Function	Node_Name	Lon_Function	SNVT_Index	SNVT_Type	Scan_Interval
nviClgSP	DA_AI_01	0	rdbc	Lon_01	NMFETCHC	2	SNVT_temp_p	1.0s
nviHtgSP	DA_AI_01	4	rdbc	Lon_01	NMFETCHC	3	SNVT_temp_p	1.0s
nviRhSP_4	DA_AI_01	13	rdbc	Lon_01	NMFETCHC	4	SNVT_lev_percent	1.0s

Device Status Monitoring

- ◆ Knowing which remote devices are communicating successfully is needed in data quality monitoring
- ◆ FieldServer can manage device status in two ways:
 1. Via Active Mappings (Automatically implemented)
 2. Via the node status function (Configurable option)
- ◆ Monitoring remote device status via active mapping can be turned off to allow for special situations.

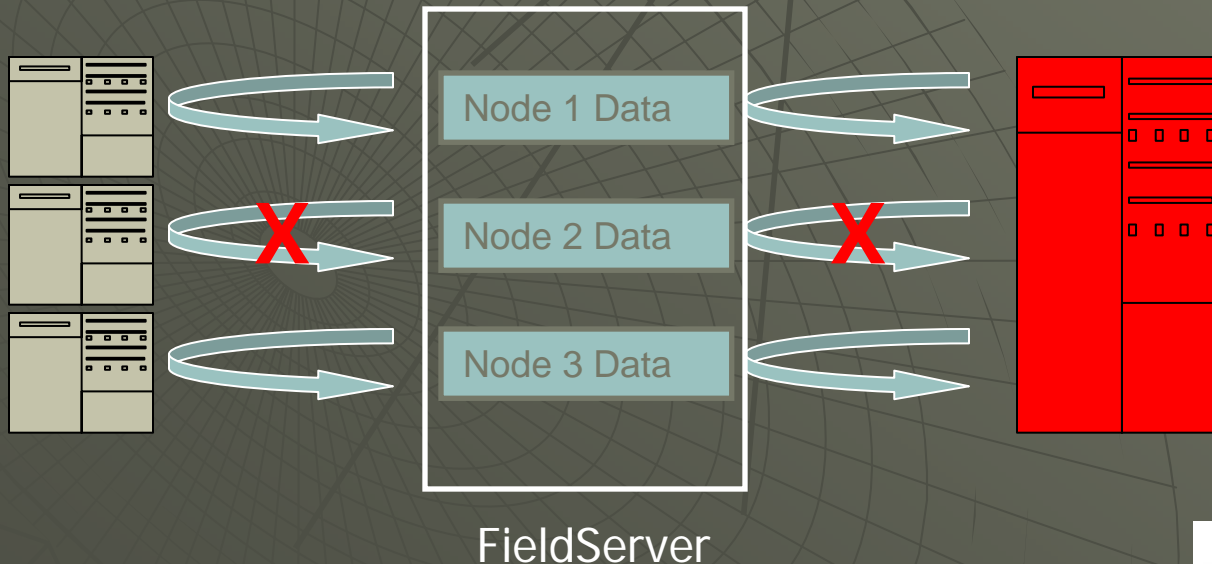
Device Status Monitoring

- ◆ FieldServer uses active mapping device status monitoring to prevent poor quality data from being propagated.
- ◆ Node Status function only provides visible device status and does not automatically control communication.

Device Management Options

Model 1: Automatic using Active Mappings

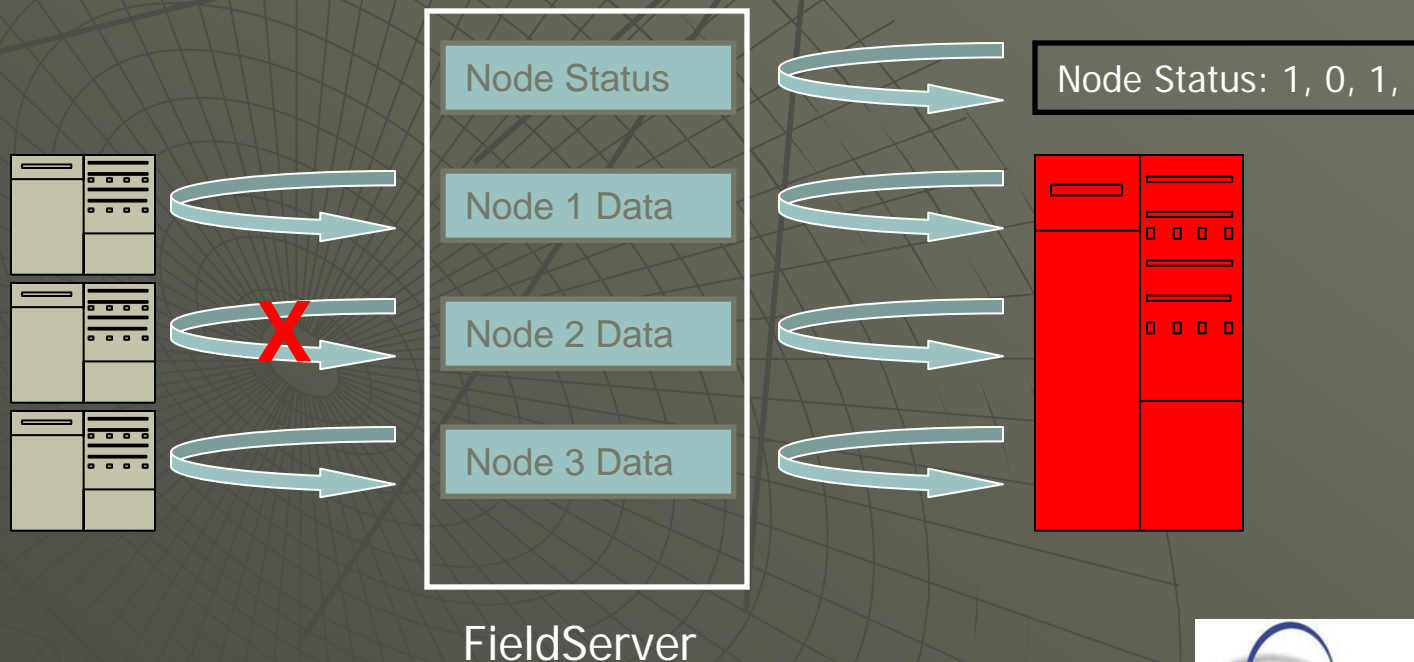
FieldServer automatically monitors data quality, and prevents propagation of poor quality data.



Device Management Options

Model 2: Configured using Node Status

Client uses Node Status bits to distinguish between good and bad quality data.



Data Typing

- ◆ Different Protocols support different data types
- ◆ All data types are variations on binary or analog data type
- ◆ Analog data types vary based on communications direction and resolution.

```
//-----  
//  
//   Data Arrays  
//  
  
Data_Arrays  
Data_Array_Name , Data_Format      , Data_Array_Length  
Status_DA      , Fl oat            , 200  
Al arm_DA      , Bi t                            , 600
```

Data Typing

- ◆ The FieldServer's challenge is to provide data storage of these varying data types that satisfies all protocols.
- ◆ FieldServer does this by allowing Data Arrays to store data in a format that is configurable.
- ◆ By default, data is typecast into the format specified for the Data Array before it is stored. This can be suppressed for special situations (e.g.: packed bit)

Data Typing

Mac's rule:

- ◆ Although FieldServer provides for many different data array formats, nearly all applications can be satisfied with a combination of Float, Bit and Packed_Bit data arrays.

Data Typing

The problem with Data Typing:

- ◆ Data quality can be compromised if data formats for transfer and storage of data are not considered.

e.g:

1. Transferring small values using integer format (eg: 1.234) creates data loss after the decimal place.
2. Transferring large values (>255) in a byte data type results in unusable data.

Timing

It is important to communicate data in real time.

So how do we define "real time"



Timing

Too many definitions for "Real Time" exist

But....

We can accept that data is real time if it is still relevant when it reaches its destination.

Read/Write Management

What's the big deal about read write management?

Simply put:

When a value is written to a remote device, and a read of the same value is executed shortly afterward, what is the value being read back?

Is it:

- 1) The current value of the point, or
- 2) The echo of the value just written.

The answer relates to timing....

Read/Write Management

- ◆ If not properly managed, read/write control can result in oscillating data or incorrect feedback of setpoints and control points.
- ◆ FieldServer has invested heavily in ensuring that read/write control is properly managed to avoid setpoint oscillation or incorrect feedback.

Resources

- ◆ FieldServer Website (www.FieldServer.com)
 - FieldServer Configuration Manual
 - Application notes (ENotes)



Questions?

Email Mac at:

mac@sfintegration.com

Email FieldServer Tech Support at:

support@fieldserver.com

Check the website for documentation:

www.fieldserver.com



THANK YOU!

.....for taking the time to attend
this presentation.