

# Microsoft RTI Migration

## Table of contents

[Fabric Workspace](#)

[Azure Data Explorer Cluster](#)

[Ingestion](#)

[Function for Ingest -> Silver Update Policy](#)

[Equipment Data Parsing](#)

[IRI - Measurements](#)

[IRI - Defects](#)

[Coldsystem - MX](#)

[Blank-Watch - Temperature](#)

[Blank-Watch - Gob-Loading](#)

[Historical Data](#)

[Source](#)

[Screenshots of Dashboards](#)

## Fabric Workspace

This is the link to the newly created Fabric Workspace we are migrating to

<https://app.powerbi.com/groups/26056bef-49f8-42ef-b0ad-faf4ae8668ea/eventhouses/0b15bedd-8523-49c3-84b1-1a136de22b1b?experience=power-bi>

## Azure Data Explorer Cluster

This is the link to the existing Azure Data Explorer Cluster w/ historical data

<https://adx-ultra-useast.eastus.kusto.windows.net>

## Ingestion

Data is ingested from the following Event Hubs

- evhns-midas-eastus-01/evh-nodered-northamerica-01
- evhns-midas-eastus-01/evh-nodered-sourthamerica-01
- evhns-midas-eastus-01/evh-nodered-europe-01

Ingest table schema is as follows:

```
1 .create table ['ingest'] (timestamp: datetime, ['data']: dynamic, properties: dynamic)
```

Silver table schema is as follow:

**Note:** for the migration, this table should be called 'bronze'

```
1 .create table silver (timestamp: datetime, device: string, category: string, ['location']:
```

```
string, productionLine: string, lineUp1: string, ['data']: dynamic)
```

## Function for Ingest -> Silver Update Policy

```
1 .create-or-alter function with (docstring = "Defines the transformation from ingest table to
  silver table",folder = "UpdatePolicies") UpdatePolicy_Ingest() {
2 ingest
3 | project timestamp, device=tostring(properties.device), category=tostring(properties.category),
  location=tostring(properties.location), productionLine=tostring(properties.productionLine),
  lineUp1=tostring(properties.lineUp1), data
4 }
```

## Equipment Data Parsing

! Note - there is no time or production line filter applied to the queries in this section.

Device and Data columns are used to filter by equipment type. For the scope of this effort, the following equipment types will be considered:

	≡ device	≡ category	≡ rows/minute/production_line
1	iri	measurements	~300 - 500
2	iri	defects	~100 or less
3	coldsystem	mx	~500 - 800
4	blank-watch	temperature	~400 - 600
5	blank-watch	gob-loading	~300 - 400

The plan for this migration is to:

1. Take the KQL queries in the following sections and turn them into functions
2. Use those functions in update policies
  - a. Trigger on bronze table
  - b. Publish to silver table(s) for each device/category

## IRI - Measurements

Proposed name for target silver table: Silver\_IRIMeasurements

```
1 bronze
2 | where device == 'iri' and category == 'measurements'
3 | project utctimestamp=timestamp, location, production_line=productionLine, lineup1=lineUp1,
  id_container=tostring(data.idContainer), leg=tostring(data.fp), pocket=tostring(data.pocket),
```

```

    cavity=toString(data.cavity), position=toString(data.position), measurements=data.measurements
4 | mv-expand kind=array measurements // New line for each inspection_name
5 | extend inspection_name=toString(measurements[0]), measurements=measurements[1]
6 | mv-expand kind=array measurements // New line for each measurement_name for a given
    inspection_name
7 | extend measurement_name = toString(measurements[0]),
    measurements=bag_remove_keys(measurements[1],dynamic(['probeNames','thresholds'])),
    probe_names=measurements[1]['probeNames'], thresholds=measurements[1]['thresholds']
8 | mv-expand kind=array measurements // New line for each measurement_probe
9 | extend probe_number=substring(measurements[0],6,1), value=toreal(measurements[1]) // Extract
    probe number to pair with label and threshold as well as measurement value
10 | extend probe_label = strcat(probe_number, " ", probe_names[probe_number]), threshold =
    thresholds[probe_number]
11 | project-away measurements, probe_names, thresholds, probe_number

```

## IRI - Defects

Proposed name for target silver table: Silver\_IRIDefects

```

1 bronze
2 | where device == 'iri' and category == 'defects'
3 | project
    utctimestamp=timestamp,lineupl=lineUp1,location,id_container=toString(data.idContainer),leg=toString(data.fp),position=toString(data.position),defects=data.defects
4 | mv-expand kind=array defects // New line for each inspection_name that recorded a defect
5 | extend inspection_name=toString(defects[0]), defects=defects[1]
6 | mv-expand probe = bag_keys(defects)
7 | extend defect_name = toString(defects[toString(probe)])
8 | project-away defects

```

## Coldsystem - MX

Proposed name for target silver table: Silver\_TESMeasurements

```

1 bronze
2 | where device == 'coldsystem' and category == 'mx' | project utctimestamp=timestamp, location,
    production_line=productionLine, lineupl=lineUp1, id_container=toString(data.ContainerID),
    leg=toString(data.Leg), pocket="", cavity=toString(data.MoldNumber),
    position=toString(data.Position), measurements=data.measurements
3 | mv-expand kind=array measurements // New line for each sensor ID
4 | extend inspection_name=toString(measurements[0]), measurements=measurements[1].measures
5 | where isnotempty(measurements) // Some sensors appear to not have a measurement
6 | mv-expand kind=array measurements // New line for each measurement_name for a given sensor ID
7 | extend probe_label = "", threshold=real(null) // Matching schema from IRI even though these do
    not exist for this machine
8 | extend measurement_name=toString(measurements[0]), value=toreal(measurements[1])
9 | project-away measurements

```

## Blank-Watch - Temperature

Proposed name for target silver table: Silver\_BlankWatchTemperature

```
1 bronze
2 | where device == 'blank-watch' and category == 'temperature'
3 | project utctimestamp=timestamp, location, production_line=productionLine, lineup1=lineUp1,
   cycle=tostring(data.cycle), section=tostring(data.section), gob=tostring(data.gob),
   temp_timestamps = parse_json(data.temp_timestamps), temperatures = parse_json(data.temperatures)
4 | mv-expand measurement_name = bag_keys(temperatures)
5 | extend temperature = todouble(temperatures[tostring(measurement)]), utctimestamp_measurement =
   todatetime(temp_timestamps[tostring(measurement)])
6 | project-away temp_timestamps, temperatures
```

## Blank-Watch - Gob-Loading

Proposed name for target silver table: Silver\_BlankWatchLoading

```
1 bronze
2 | where device == 'blank-watch' and category == 'gob-loading'
3 | project utctimestamp=timestamp, location, production_line=productionLine, lineup1=lineUp1,
   cycle=tostring(data.cycle), section=tostring(data.section), gob=tostring(data.gob),
   position=tostring(data.position),
   data=bag_remove_keys(data,dynamic(['position','cycle','section','gob']))
4 | mv-expand kind=array data
5 | extend measurement_name=tostring(data[0]), value=toreal(data[1])
6 | project-away data
```

## Historical Data

### Source

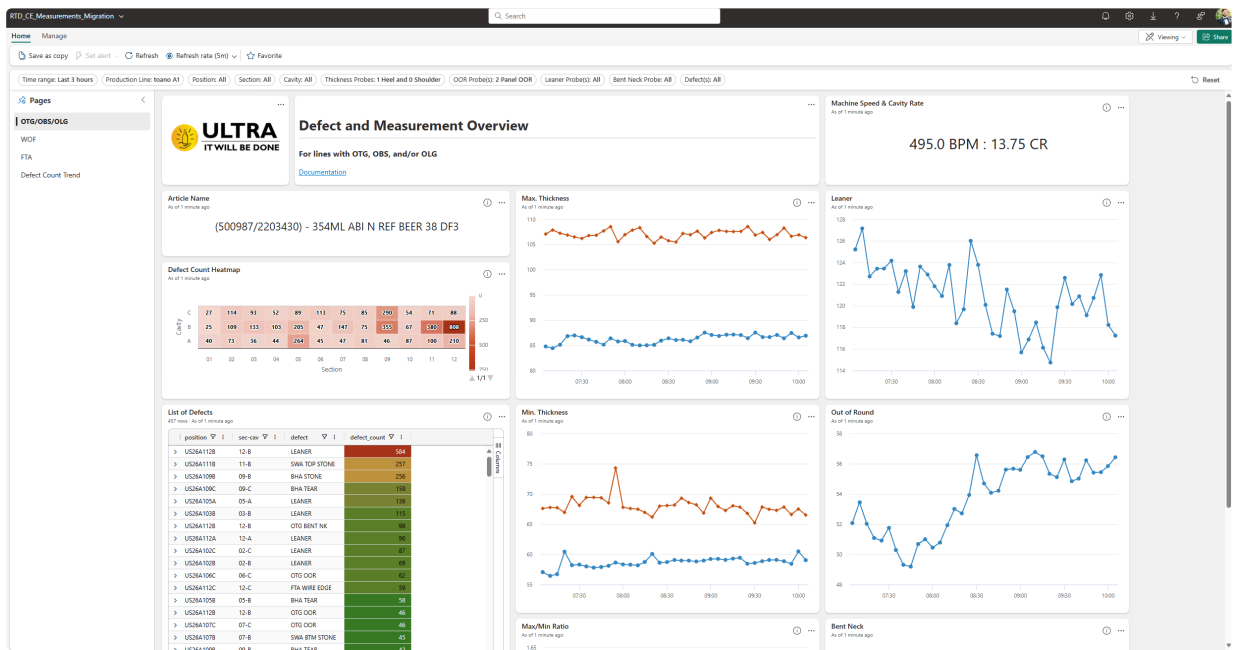
Historical data to be ingested exists in the standalone ADX Cluster's Silver Table.

-> ADX Silver Table inputs to Fabric Eventhouse Bronze Table

## Screenshots of Dashboards

Visualization for IRI - Measurements and IRI - Defects

RTD\_CE\_Measurements



Visualization for Blank-Watch - Temperatures and Blank-Watch - Loading

Note: This dashboard was put together for troubleshooting as there is no standard report built yet

RTD\_HE\_BlankWatch

