Modernizing PI SQL For The Future

Bodo Bachmann
Christopher Sawyer

Your Speakers Today

- Bodo Bachmann
- Team Leader, Software Development
- bodo@osisoft.com

- Christopher Sawyer
- Technology Enablement
- csawyer@osisoft.com

Why SQL for the PI System

- Simplest place to cross-pollinate actionable line-of-business (LOB) data with real-time data
- Enhancing Real Time Data with LOB data
- Enhancing LOB Data with RT
- Maximizes the resources you already have to complete data projects:
  - SQL Database Administrators
  - Web Developers
  - LOB App Developers

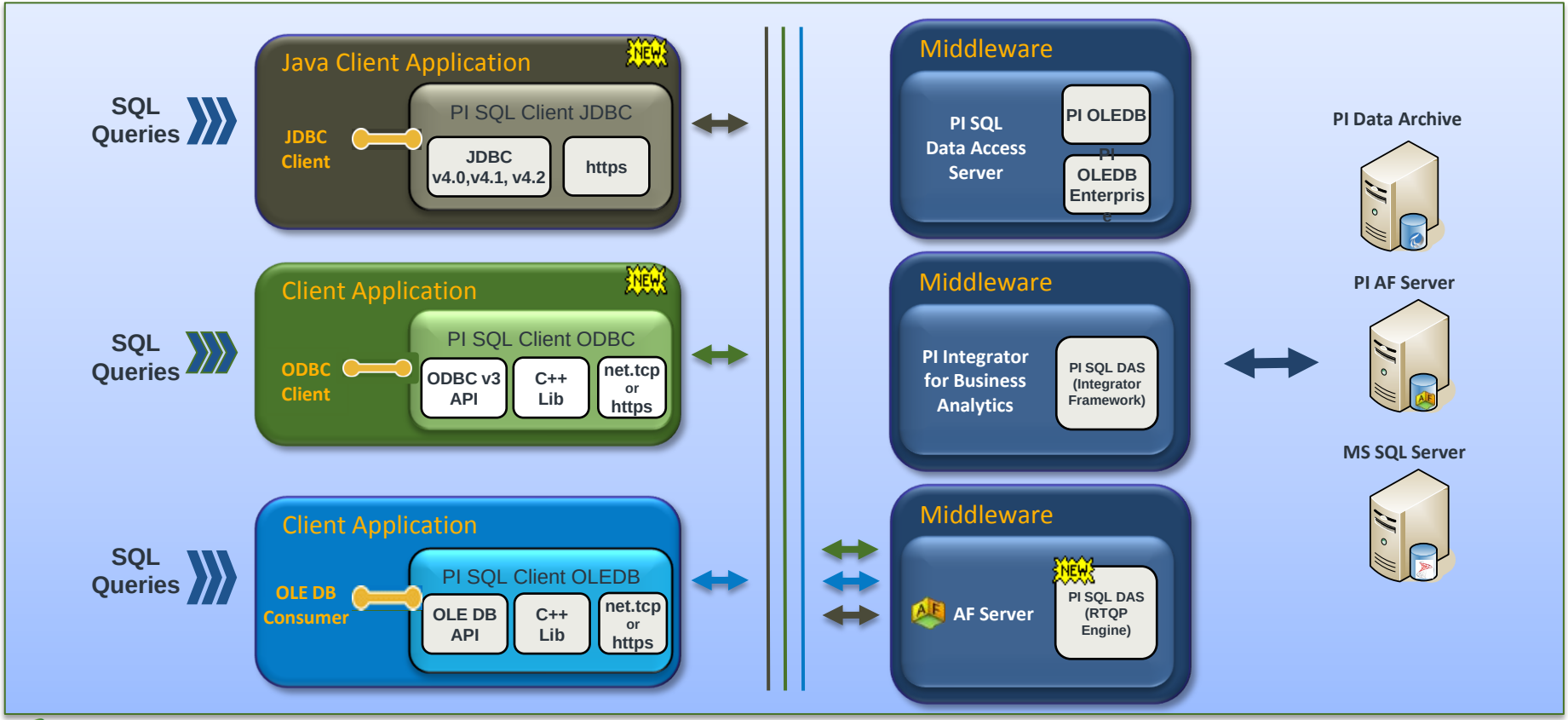
Why SQL for the PI System

- Some data will never live in SQL (streams)
- Some data will never live in the PI System (financial costing)
- We've optimized how we merge relational structures and their relationships to assets with streaming data in the PI System

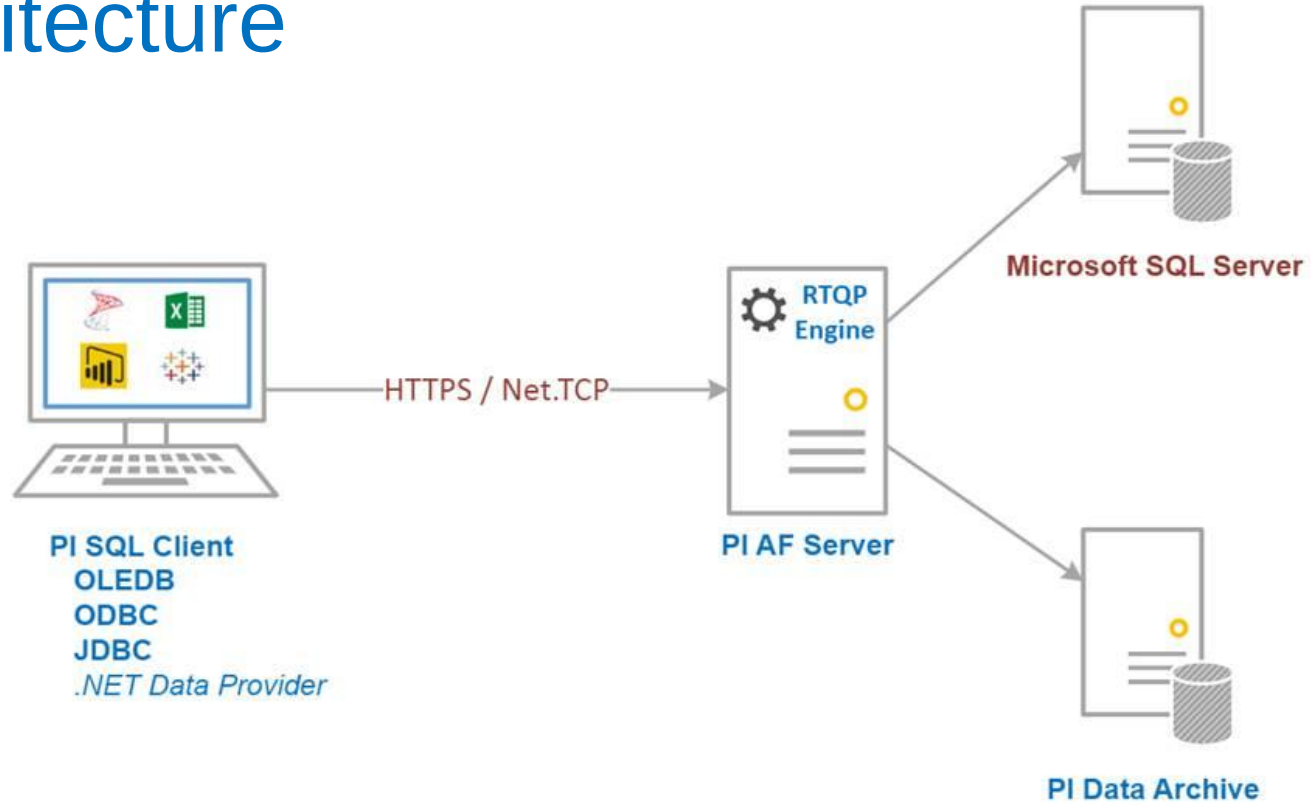
Next Generation Development Goals

- Performance and scalability
- Review data model
  - Reduce query complexity
  - What are top use cases (use customer feedback)
  - Address known issues (e.g. timestep data type, PI time literals)
- Infrastructure to support multiple standards and thin clients
- Works across WAN and different time zones
- Review security and authentication

PI SQL Client Architecture



Architecture



RTQP Engine – A Learned Technology

- **Deployment**
- **Working with different internal data sources**
- **Issue of Data References**
- **Memory Management / Intermediate Results**
- **Optimization**
- **Streaming**
- **Security**

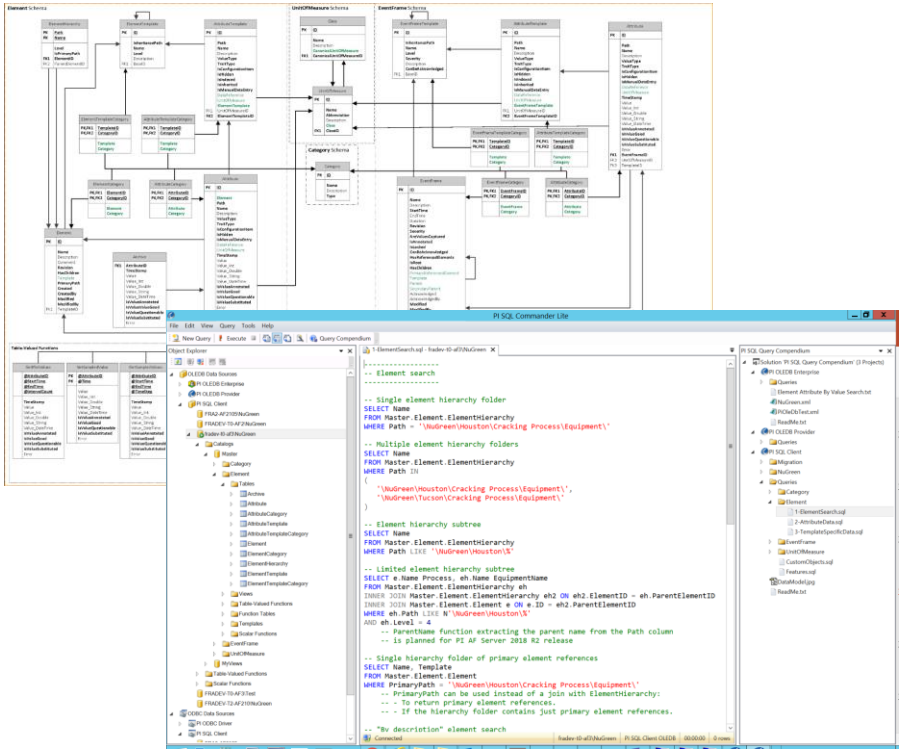
RTQP Engine: Data Model

- Entity-Relationship Model

- Connection = AF Database
- Master Catalog

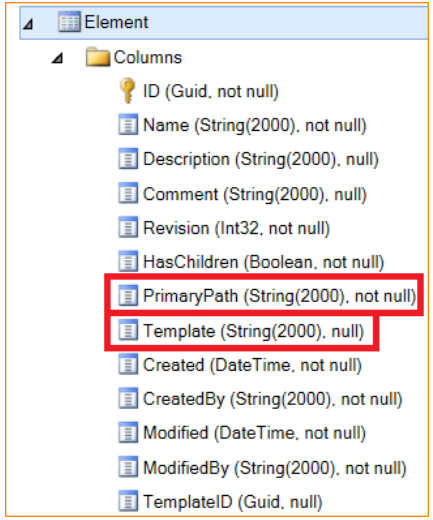
- Elements and Event Frames
- Attributes and Data
- Categories and Templates
- Template based Table-Valued Functions (*New!*)

- Denormalization
- Navigation Columns
- Multiple Value columns



Simplified Queries

Denormalization

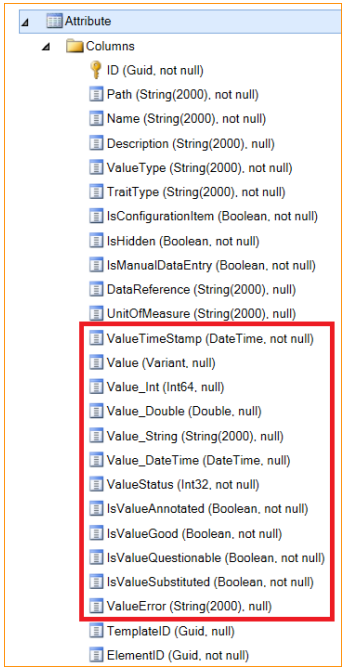



```
SELECT eh.Path + eh.Name Element, et.Name Template
FROM NuGreen.Asset.Element e
LEFT OUTER JOIN NuGreen.Asset.ElementTemplate et
    ON et.ID = e.ElementTemplateID
INNER JOIN NuGreen.Asset.ElementHierarchy eh
    ON eh.ElementID = e.ID
WHERE et.Name IN ('Boiler', 'Heater')
```

```
SELECT e.PrimaryPath + e.Name, e.Template
FROM Master.Element.Element e
WHERE e.Template IN ('Boiler', 'Heater')
```

Simplified Queries

Extention




```
SELECT eh.Path + eh.Name Element, et.Name Template, ea.Name Attribute, s.Time, s.Value
FROM NuGreen.Asset.Element e
LEFT OUTER JOIN NuGreen.Asset.ElementTemplate et    ON et.ID = e.ElementTemplateID
INNER JOIN NuGreen.Asset.ElementHierarchy eh    ON e.ID = eh.ElementID
INNER JOIN NuGreen.Asset.ElementAttribute ea    ON eh.ElementID = ea.ElementID
INNER JOIN NuGreen.Data.Snapshot s              ON ea.ID = s.ElementAttributeID
WHERE et.Name IN ('Boiler', 'Heater')
      AND eh.Path = N'\NuGreen\Houston\Cracking Process\Equipment\'
ORDER BY 1
```

```
SELECT e.PrimaryPath + e.Name Element, e.Template, ea.Name Attribute,
       ea.ValueTimeStamp, ea.Value
FROM Master.Element.Element e
INNER JOIN Master.Element.Attribute ea     ON e.ID = ea.ElementID
WHERE e.Template IN ('Boiler', 'Heater')
      AND e.PrimaryPath = N'\NuGreen\Houston\Cracking Process\Equipment\'
ORDER BY 1
```

RTQP Engine: Custom Objects

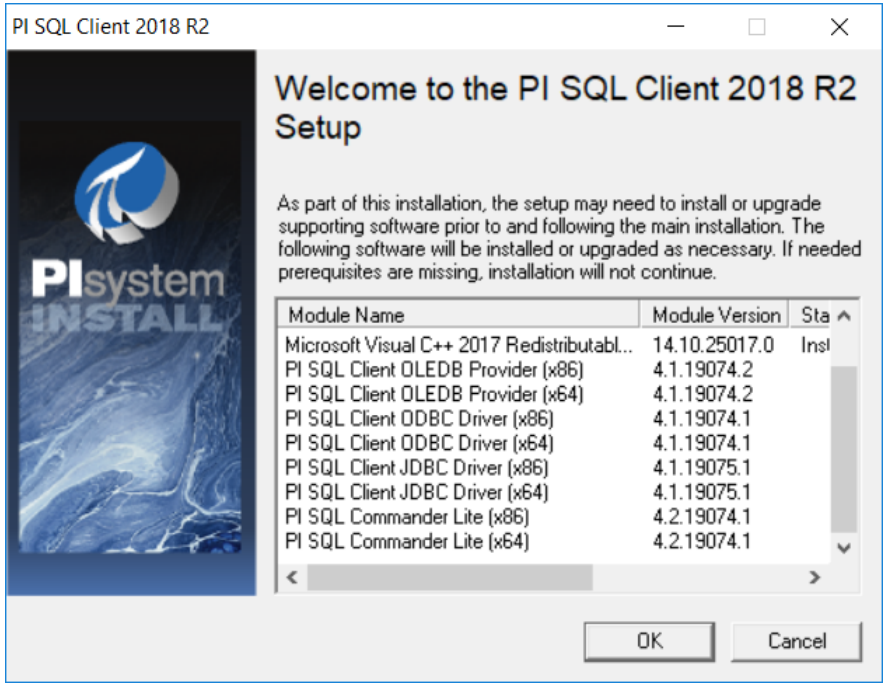
- Views
- Custom Table-Valued Functions
- Function Tables
  for Table-Valued Functions
- Custom Functions

```
-- Create a custom catalog
CREATE CATALOG Test

-- Create a custom schema
CREATE SCHEMA Test.Test

-- Create a custom view
CREATE VIEW Test.Test.HoustonElement
AS
SELECT *
FROM Master.Element.Element
WHERE PrimaryPath LIKE '\NuGreen\Houston\%'


-- Create a custom table-valued function as a parameterized query
-- (all parameters must be used in WHERE clause column equalities)
CREATE FUNCTION Test.Test.YesterdayEventFrame(@Unit String)
AS
SELECT ef.Name, ef.StartTime, ef.Duration, sv.Value ProcessFeedrate
FROM Master.EventFrame.EventFrame ef
INNER JOIN Master.Element.Attribute a ON a.ElementID = ef.PrimaryReferencedElementID
CROSS APPLY Master.Element.GetSampledValue(a.ID, ef.EndTime) sv
WHERE ef.PrimaryReferencedElement = @Unit
AND a.Name = 'Process Feedrate'
```

SQL Clients

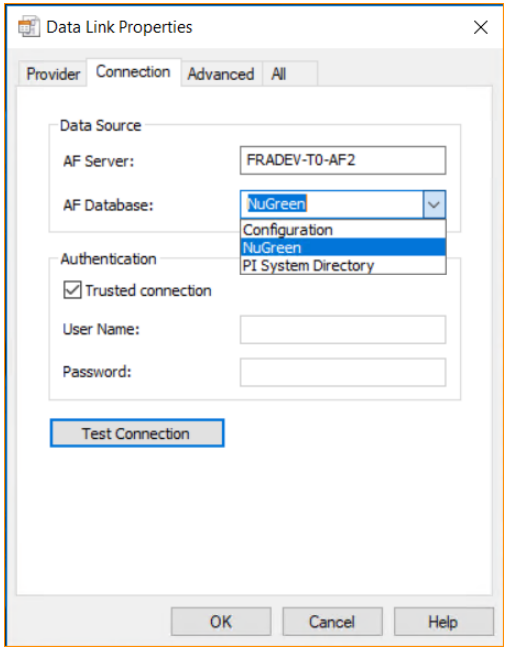
PI SQL Client 2018 R2

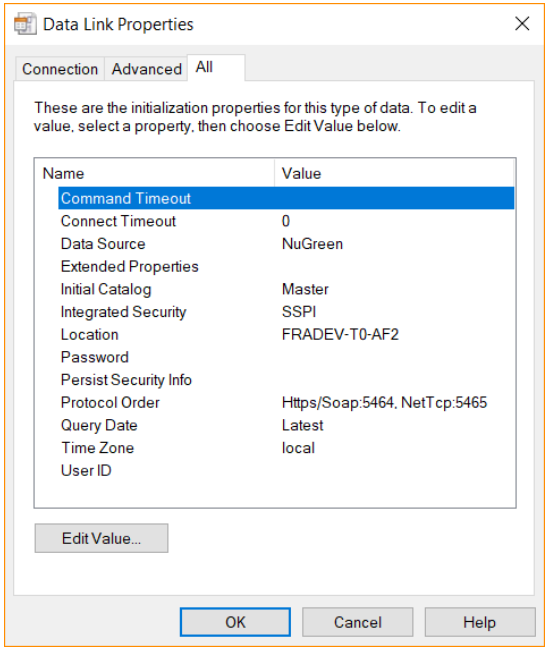
- No AF SDK dependency
- ODBC added
- JDBC added



Connection and Connection Options

- AF Database names can be browsed when PI Web API Core Services are configured
- Time Zone support
- Query Date





PI SQL Client: OLE DB Provider

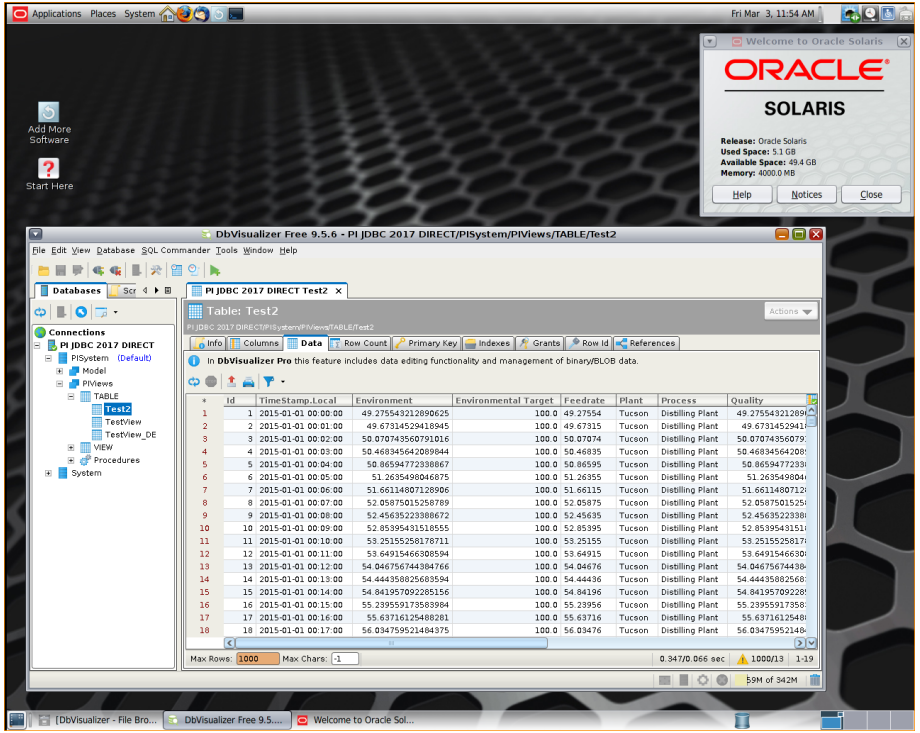
- SQL Server integration (SSRS, SSIS, etc.)
- BizTalk
- SAP Crystal Reports
- ADO.NET, ASP.NET

PI SQL Client: ODBC Driver

- Oracle integration
- BI Tools, e.g. Tibco, Spotfire, Power BI
- Crystal Reports
- ADO.NET, ASP.NET

PI SQL Client: JDBC Driver

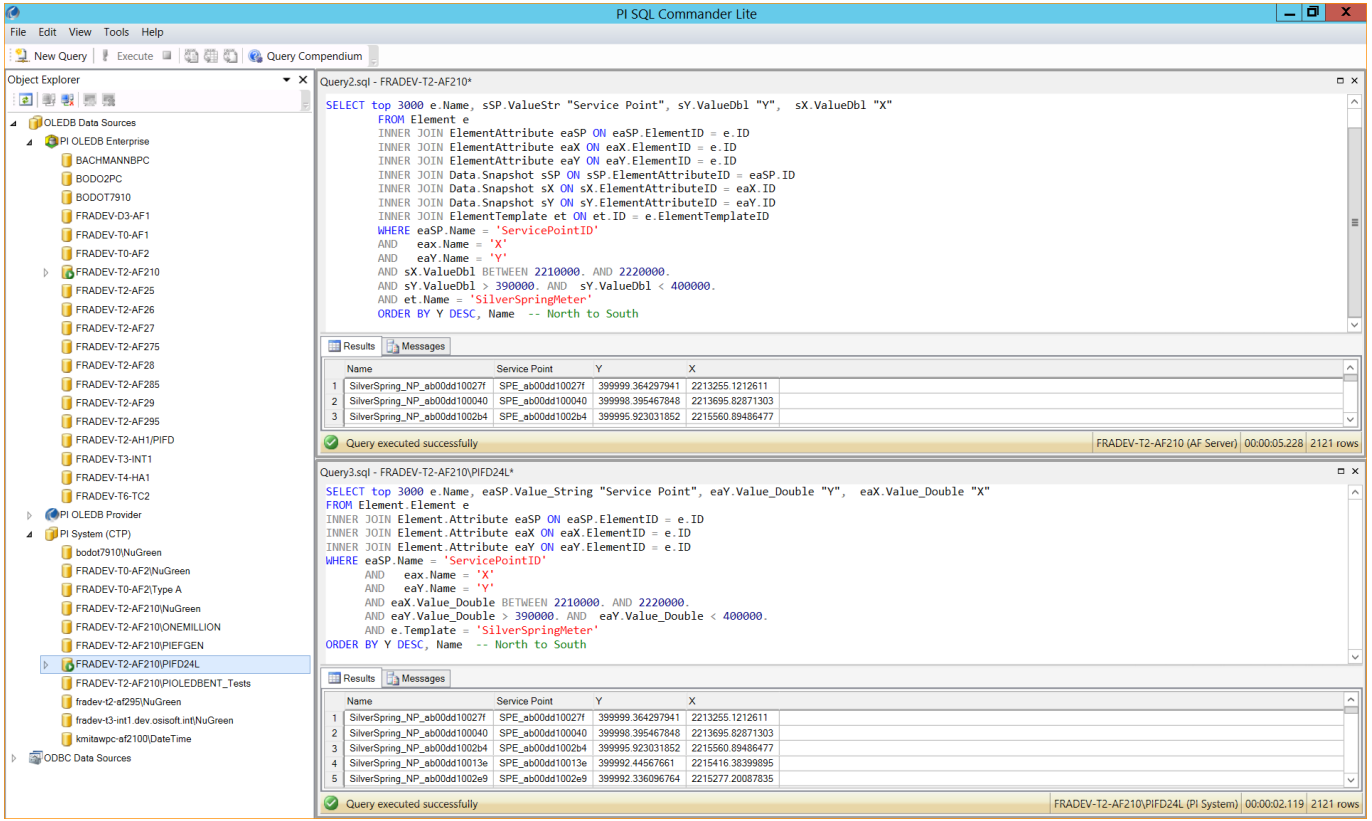
- Pure Java
- Https only



PI SQL Client: PI SQL Commander

- Browse Data Model and Functions
- Compendium
- Code/keyboard pre-completion (it might also be known as IntelliSense™®©☺)

Sample Queries and Performance

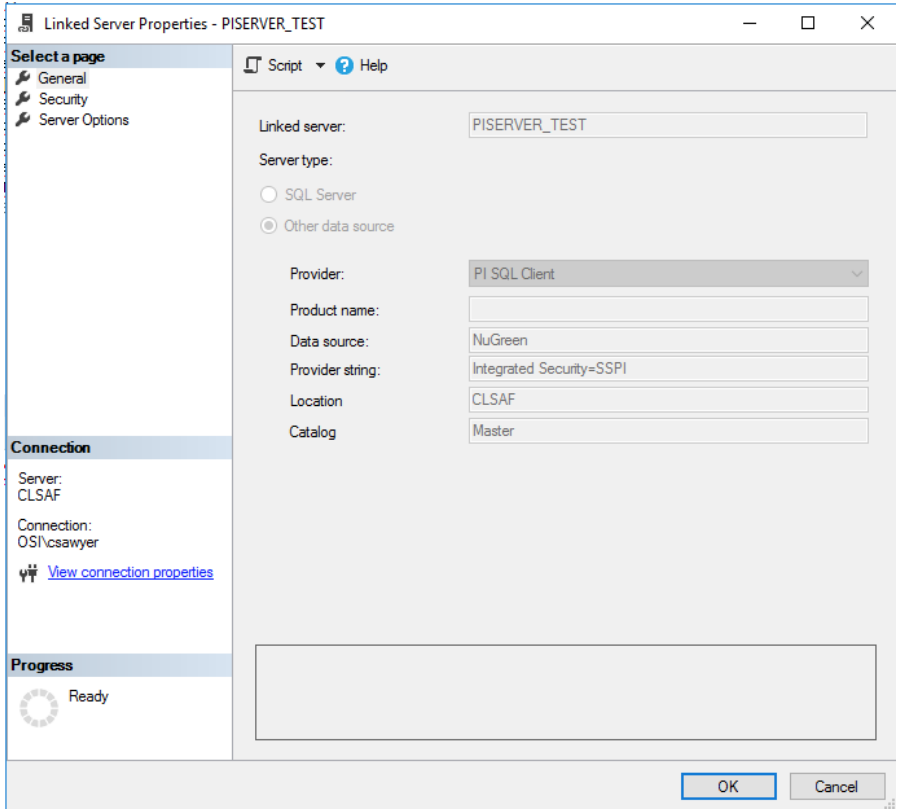


Future

- Locale property:
  - e.g. '01-Dec-2019' vs N'01-Dez-2019’
- Data Archive tables (PI tags vs AF attributes)
- INSERT/UPDATE/DELETE event frames
- INSERT/UPDATE/DELETE tags/attribute values

DEMO

Microsoft SQL Server – PI SQL Client Remote Connection



Questions?

Please wait for the **microphone**

State your **name & company**

Please remember