



PlantPax Reporting

Topic	Page
Before You Begin	5
PlantPax Centralized Database	17
Rockwell Automation Standard Reports	33
ThingWorx Foundation	37
ThingWorx Maintenance Reports	47
Control Strategy Health Dashboards	53
Vuforia Experiences	65
Event Queue Solution	73
Troubleshooting	117



About This Publication

This application technique covers Reporting and Maintenance capabilities that are introduced with the PlantPax® 5.0 System Release. Installation and configuration details are included for the following:

- Rockwell Automation Standard Reports
- ThingWorx® Maintenance Reports
- ThingWorx Control Strategy Health Dashboards
- Vuforia® Augmented Reality (AR) Experiences
- PlantPax EventQ Solution

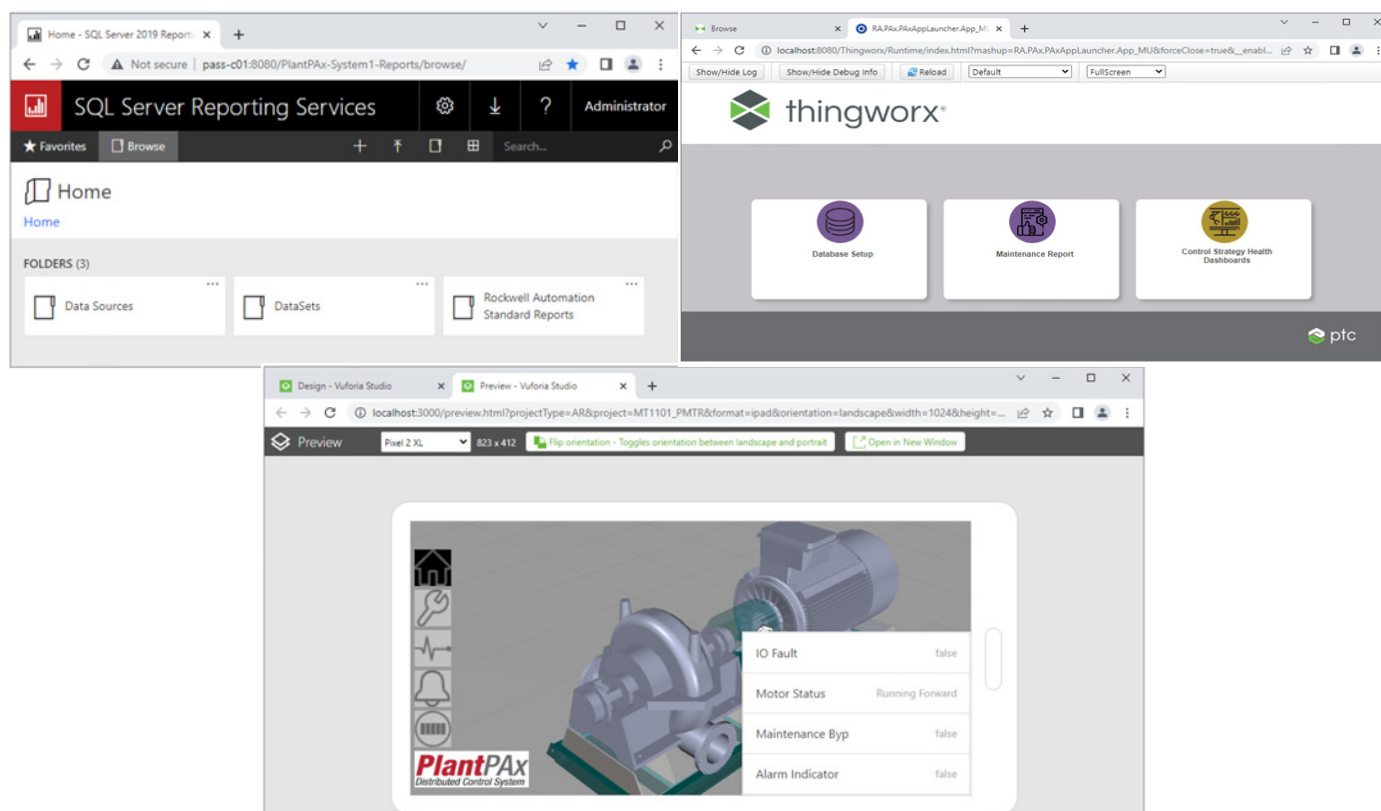
Introduction

The PlantPax 5.0 Distributed Control System provides users with tools and templates to accelerate your organization's digital transformation. New reporting tools collect and enrich control system data. Plant operators and managers can use these reports to gain valuable insights into process operations.

There are three available report classes: Alarms & Events, Process Object, and Traceability and System.

- Alarms & Events – Area-based or Object-based. Controller and Server alarms. Follows ISA 18.2 guidelines for chattering, fleeting, flooding, and stale alarms.
- Process Object – Control module information. Captures additional insights by analyzing historical data. Monthly, Daily, and Shift reports.
- Traceability and System – Audit Trails. Highlight objects with excessive alarms. Provide state of reporting systems.

This document helps you implement different analytical capabilities that are provided with the PlantPax Process Library. When an issue arises in your plant, you can perform root cause analysis by using SSRS standard reports, ThingWorx maintenance reports, and the live-data Control Strategy Health Dashboards. Further solutions can be built with Thing Templates, such as our Vuforia experiences.



Notes:

Before You Begin

PlantPax 5.0 System

This document assumes that the PlantPax® 5.0 System has already been installed and configured in terms of:

- Process Controllers contain running applications
- FactoryTalk® View SE server running an HMI application
- FactoryTalk Alarms & Events server with SQL logging
- FactoryTalk AssetCentre server with SQL logging
- SQL Server Standard Edition Database
- FactoryTalk Historian SE server
- FactoryTalk Historian Asset Framework Database
- PTC® ThingWorx® Foundation server (Optional: Not required for SSRS reports)

Installation and Configuration instructions for a PlantPax system are found in the following publications.

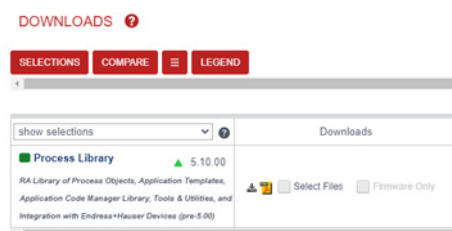
Publication	Notes
PlantPax DCS - Configuration and Implementation Manual, Publication PROCES-UM100	Provides system guidelines and instructions to assist with the development of your PlantPax system. <ul style="list-style-type: none"> • Process Automation System Server chapter <ul style="list-style-type: none"> - Add the Alarms and Events Database section - Add a Data Server section - Add an Alarm Server section Ensure you enable historization on both the Data Server and the Alarm Server <ul style="list-style-type: none"> • Historical Data chapter.
PlantPax DCS - Template User Manual, Publication 9528-UM001	Provides direction on how to install and deploy PlantPax virtual templates. NOTE: Only needed if you have purchased the PlantPax Virtual Image Templates. <ul style="list-style-type: none"> • Configure the SQL Virtual Server chapter • Configure the Asset Management Virtual Server chapter
PTC ThingWorx Platform 9 Help Center	Provides direction on how to install ThingWorx Foundation for Windows®, which Apache Tomcat and Oracle Java installed. PostgreSQL, InfluxDB, or MSSQL Server.
Vuforia® Studio 9.9.0 Help Center	Provides direction on how to install Vuforia Studio.
Vuforia 9.9.0 Experience Service Help Center	The experience service for Vuforia Studio augments.

Library Downloads

Process Library

The Rockwell Automation Process Library is a predefined library of controller code (control strategies), HMI display elements (global objects), graphical faceplates, tools, and templates that let you quickly assemble large applications with proven strategies, rich functionality, and known performance.

1. For reporting templates, download the latest Process Library from Rockwell Automation's [Product Compatibility & Download Center](#).

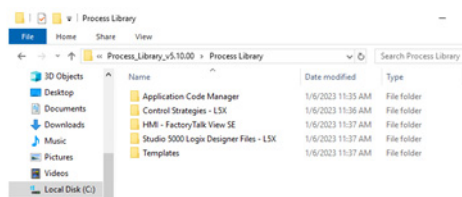


2. Extract the download library file.

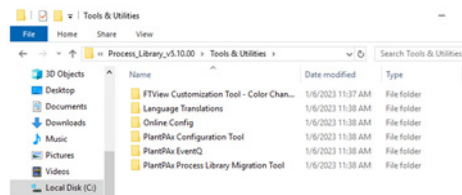


Copy the library folder to a shared network directory or copy to an engineering workstation and SQL server to ease installation within each reporting component.

3. Explore the Process Library content. The Templates directory contains the PlantPax reporting content.



4. Explore the Tools & Utilities content.

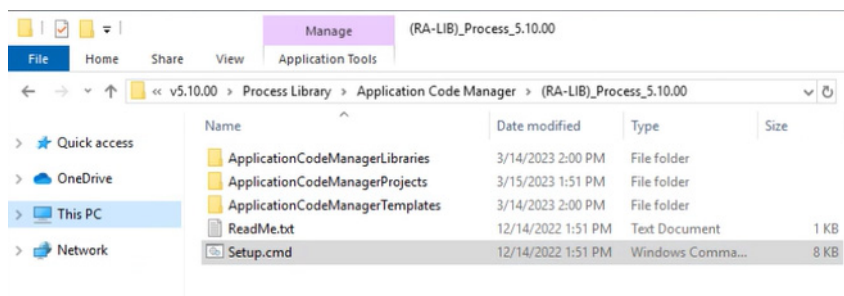


Application Code Manager (ACM) Library

ACM software is the preferred method for developing PlantPax applications and generating application content.

If you use ACM software and your PlantPax system contains a mix of process library versions, confirm that the ACM software contains the registered libraries from each version used. For example, download and register ACM library support for v5.00.00 and later libraries.

Use the Setup.cmd command script to quickly setup Rockwell Automation ACM library content.



Process Controller Applications

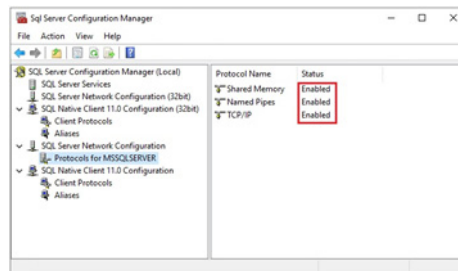
PlantPax reporting is designed for use with ControlLogix® 5580P and CompactLogix™ 5380P Process controllers.

The process controllers support the Process Library 5.0 that features firmware embedded process objects. Most of the PlantPax reporting templates are designed for the 5.0 process objects.

SQL Server Prerequisites

PlantPAx system applications require a SQL Server database for logging system data. Prerequisites for PlantPAx reporting include:

- The FactoryTalk Alarms and Events, FactoryTalk AssetCentre, and Event Queue databases must be on the same SQL server and instance.
- The Historian Asset Framework database can exist on the same SQL server or separate server.
- Standard Edition licensing is required for SQL Server Agent functionality.
- SQL Server Management Studio software installed.
- Incoming Windows Firewall rule to allow TCP Port 1433 and UDP Port 1434.
- SQL Server communication protocols must be enabled.
 - a. Open the SQL Server Configuration Manager on the SQL Server.
 - b. Expand SQL Server Network/Client Configurations > Protocols for [Instance name]
 - c. Make sure that Shared Memory, Named Pipes, and TCP/IP are enabled.
 - d. Restart SQL Server Services to apply the changes.



SQL Server uses extra memory for jobs and stored procedures but does not release it afterwards. It is recommended to set the maximum server memory in the SQL server properties to a value that reserves some memory headroom for other applications on the computer.

Install SQL Server Reporting Services

SQL Server Reporting Services provides a set of on-premises tools and services that create, deploy, and manage paginated reports. SSRS can be used with SQL Server Standard Edition. There are two methods of installing SSRS.



If using the PlantPAx AppServ-SQL Virtual Template, SQL reporting services is preinstalled.

SQL Server 2016 and Prior

- Install fundamental mode SSRS with the SQL Server installation wizard.

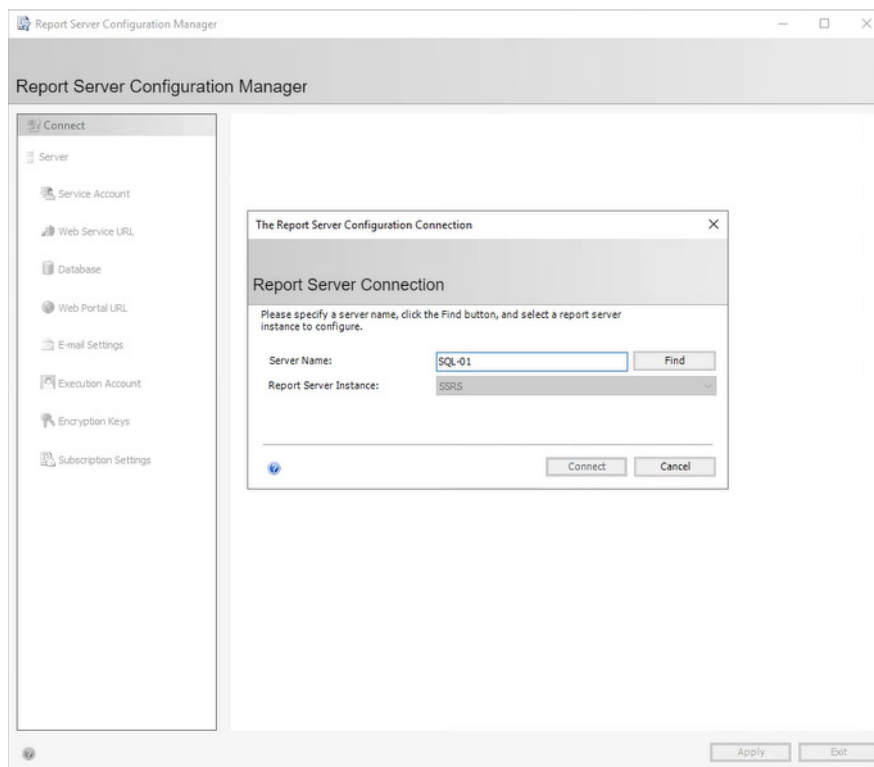
SQL Server 2017 and Later

- Download SSRS from the Microsoft® Download Center and install SSRS from the separate executable.

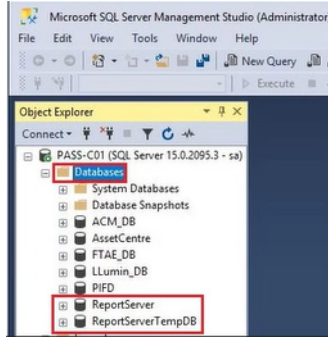
Configure SQL Server Reporting Services

This section guides you in deploying the standard reports on a computer with SQL Server 2019. All required software must already be installed.

1. From the Windows Start menu, open "Reporting Services Configuration Manager".
2. Select the SQL Server computer name and instance (should be default) and select "Connect".



3. Follow the configuration steps as described in the following table.

From Location	Action
Service Account	<p>Specify a built-in account or Windows domain user account to run the report server service.</p> <ul style="list-style-type: none"> Virtual Service Account represents the Windows service. It is a built-in least-privilege account that has network log on permissions. This account is recommended if you don't have a domain user account available or if you want to avoid any service disruptions that might occur as a result of password expiration policies. If you have a Windows domain user account that has the minimum permissions that are required for report server operations, you should use it. A domain user account is recommended because it isolates the Report Server service from other applications. <p>Avoid using a local Windows user account. Local accounts typically don't have sufficient permission to access resources on other computers.</p>
Web Service URL	<p>Configure a URL used to access the Report Server.</p> <p>A report server URL includes a virtual directory that corresponds to the application that gets the request. You must specify unique virtual directory names for each Reporting Services application that listens on the same IP address and port.</p> <p>Use the default Virtual Directory name 'ReportServer' if it's not already in use.</p> <p>Change the TCP Port to your application-specific setting, select 'Apply'. Port number 8080 is commonly used for web servers.</p> <ul style="list-style-type: none"> Create an Incoming Windows Firewall rule to allow the selected TCP Port
Database	<p>The report server stores all report server content and application data in a database. Use this page to create or change the report server database or update database connection credentials.</p> <ol style="list-style-type: none"> 1. Select the 'Change Database' button. 2. For a new installation, select to 'Create a report server database.' and select 'Next'. To connect to the Database Server, confirm that the SQL Server computer name is entered and adequate authentication is entered to create databases. Test the connection. When successful, select 'Next'. 3. Verify the default database name 'ReportServer' and select 'Next'. 4. Specify the credentials that the report server uses to connect to the report server database. For Credential type, you can use the service accounts (SA), a Windows domain account, or a SQL Server database login. The credentials that you provide must be granted access to the report server database. 5. Verify the new database account information and select 'Next'. 6. Wait for the creation process to complete, then select 'Finish' after the creation process has completed. 7. Select the 'Apply' button before continuing to the Web Portal URL configuration. <p>Microsoft SQL Server Management Studio software can be used to verify 'ReportServer' and 'ReportServerTempDB' databases have been created.</p> 
Web Portal URL	<p>Configure a URL to access the Web Portal. On a new installation, only administrators have sufficient permissions to work with content and settings. This is the URL that is used to access the reports.</p> <p>In the address bar of the web browser, type the web portal URL.</p> <p>The report server for a PlantPAx system would use a specific port and descriptive path for identification.</p> <p>For example, http://[ComputerName]:8080/PlantPAx-System1-Reports or http://[ComputerName]:8080/PlantPAx-PlantX-Reports</p>
Email Settings	Not Required
Execution Account	<p>Specify this account to enable the use of report data sources that do not require credentials or to connect to remote servers that store external images that are used in reports.</p> <p>For example, Domain\ReportUser</p>
Encryption Keys	Not Required
Subscription Settings	Not Required
Scale-out Deployment	Not Required
Power BI Service (cloud)	Not Required

Building SSRS Reports (Optional)

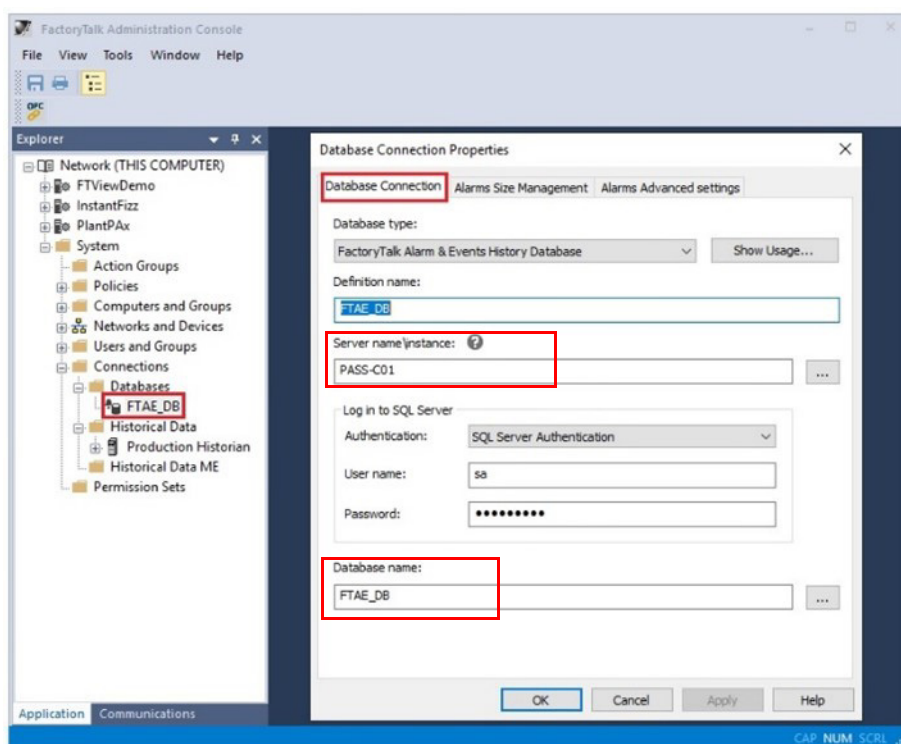
SSRS reporting webpages are built using Microsoft Report Builder or Microsoft Visual Studio®. These software packages are optional and can be used to create your own customer facing reports that can be integrated into the default dashboards to bolster your system reports.

FactoryTalk Alarms and Events

PlantPAx reporting is designed for use with a FactoryTalk View Alarms and Events server. The HMI application contains FactoryTalk Linx data servers that log Alarms and Events to a SQL database.

The Alarms and Events SQL database is created and defined using the FactoryTalk Administration Console.

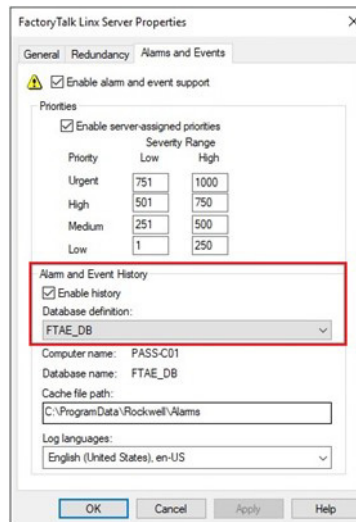
1. Launch the FactoryTalk Administration Console.
2. Expand Systems > Connections > Databases.
3. Verify the Server name\instance that is populated is the computer that you are using for PlantPAx Reporting.
4. Record the name of the Alarms and Events database for use during PlantPAx Reporting installation.



FactoryTalk View SE

PlantPAx reporting is designed for use with a FactoryTalk View SE HMI server. The FactoryTalk Linx data servers that are defined within the application require Alarm and Events support and logging configured to a SQL database.

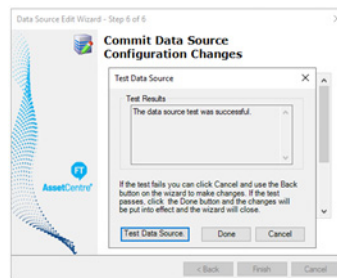
1. Confirm that your FactoryTalk Linx data servers are logging to your FactoryTalk Alarms and Events SQL database.
2. Launch the FactoryTalk View Studio software to open your HMI application.
3. Access the FactoryTalk Linx Server properties and select the “Alarms and Events” tab.
4. Verify that “Enable history” is selected and the Database definition matches what is in the [FactoryTalk Alarms and Events](#) section.



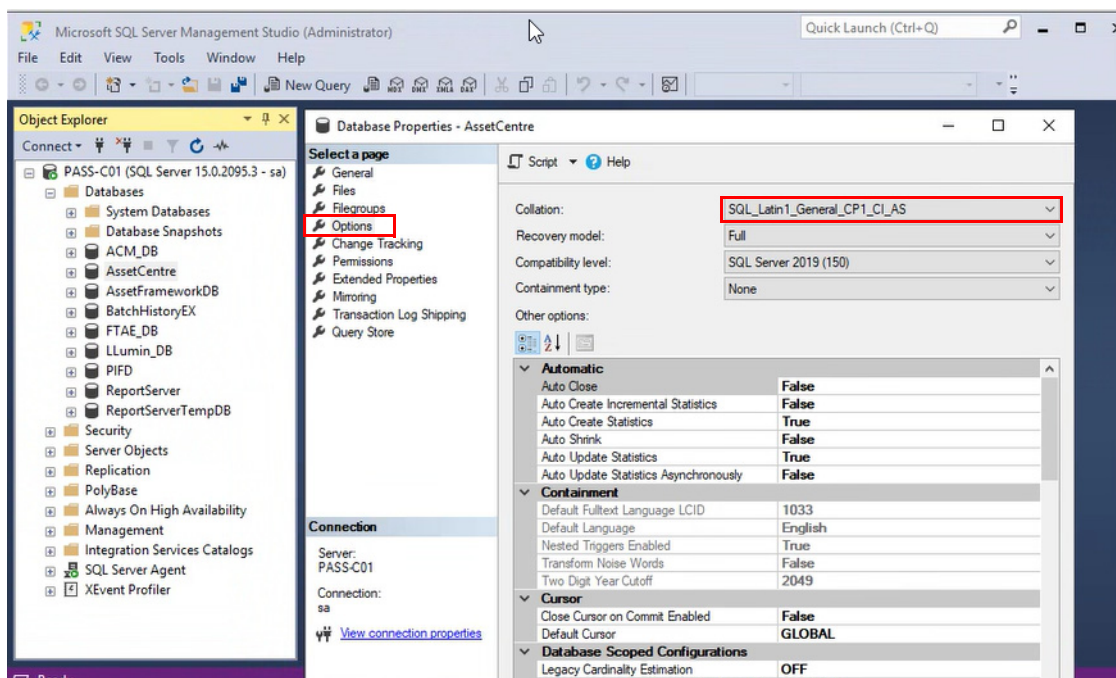
FactoryTalk AssetCentre

PlantPAx reporting is designed for use with FactoryTalk AssetCentre for traceability and system information.

1. Record the name of the AssetCentre database for use during PlantPAx Reporting installation.
2. If you do not know the name, use the Data Source Configuration wizard found on the Assetcentre Server to review the settings and test the connection to its SQL database.



- Use Microsoft SQL Server Management Studio to verify the AssetCentre database collation. Right-click the database for properties. Select the Options page and make sure that SQL_Latin1_General_CP1_CI_AS is selected.

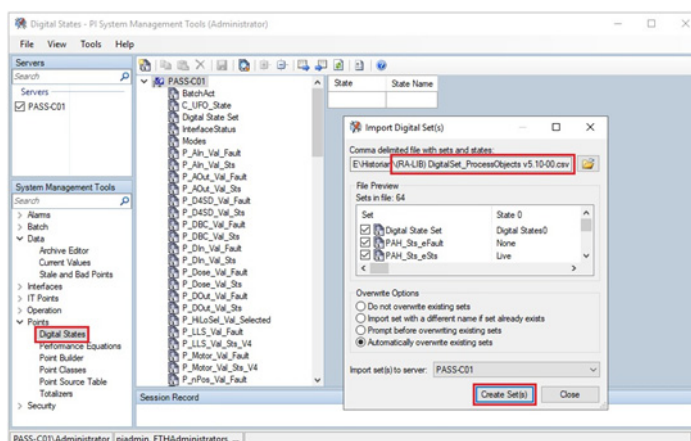


FactoryTalk Historian SE

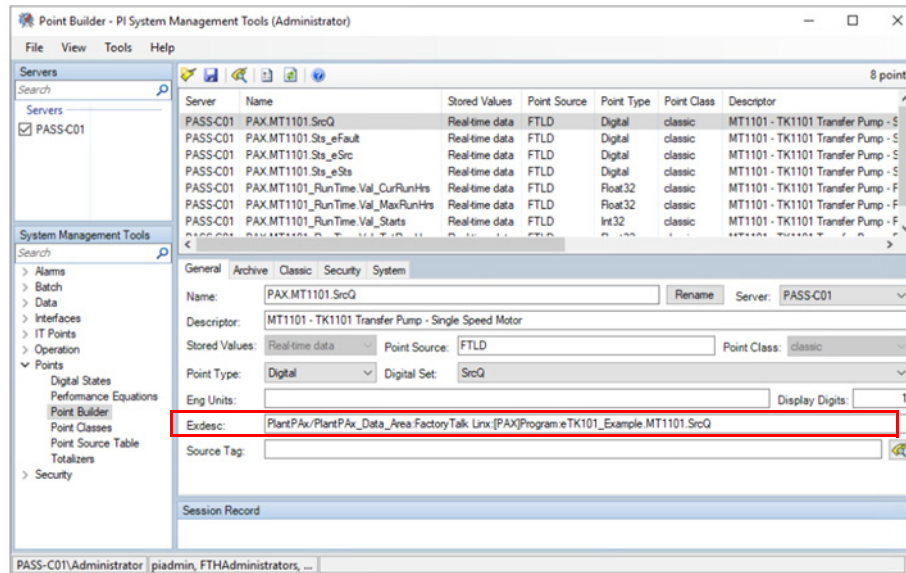
PlantPax reporting is designed for use with a FactoryTalk Historian SE server. FactoryTalk Historian software uses historical points (tags) in the system to produce analytical data for reporting.

A Process Objects Digital Set is available from the PlantPax Process Library.

- Launch the PI System Management Tools software and verify that the Process Objects digital set has been imported.
- If the digital sets have not been imported or you want to add new objects from the latest library, right-click Historian server name and select "Import from File".
- Browse the Process Library > Templates > FactoryTalk Historian SE > Historian directory for the digital set file "(RA-LIB) DigitalSet_ProcessObjects v5.xx.xx.csv".



- Next select Point Builder in the left window. Select the “Search” icon above and type the name of a few process object tags in your Historian database.



- Verify your PIPoints contain an extended description (Exdesc) on the General tab. This description would be the same path that is found for Instrument Tag on the Classic tab. Extended descriptions are required due to a limitation in Asset Framework and used to determine the controller and program of the object/element.
- If you suspect your Historian database is missing the extended descriptions, use the PI Builder Add-On in Excel® to modify your process object tags.

Historian OLEDB Connectivity

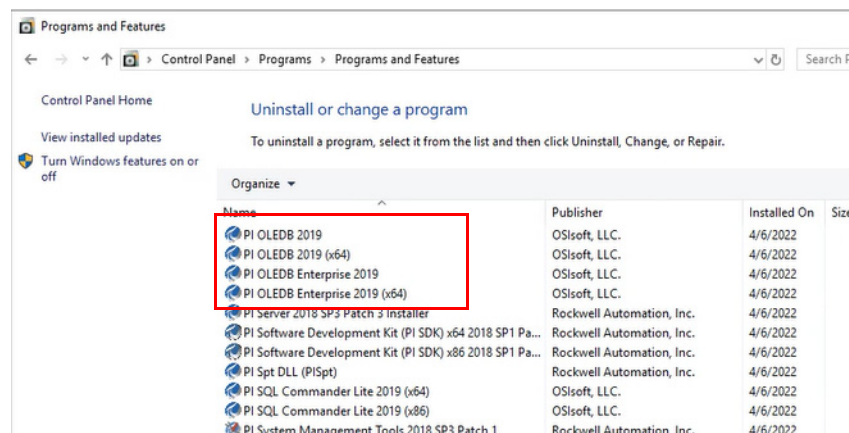
PI OLEDB Provider and PI OLEDB Enterprise components (additional activation required) provide access to the Historian system in a relational view, accessible through SQL queries. An additional license is required for the PI OLEDB software.

- PI OLEDB Provider (Classic) is used to access PI Data Archive.
- PI OLEDB Enterprise is used to access PI Asset Framework.

The PlantPax Standard Reports script will help users to create Linked Server connections to both the Asset Framework database and the Historian database.

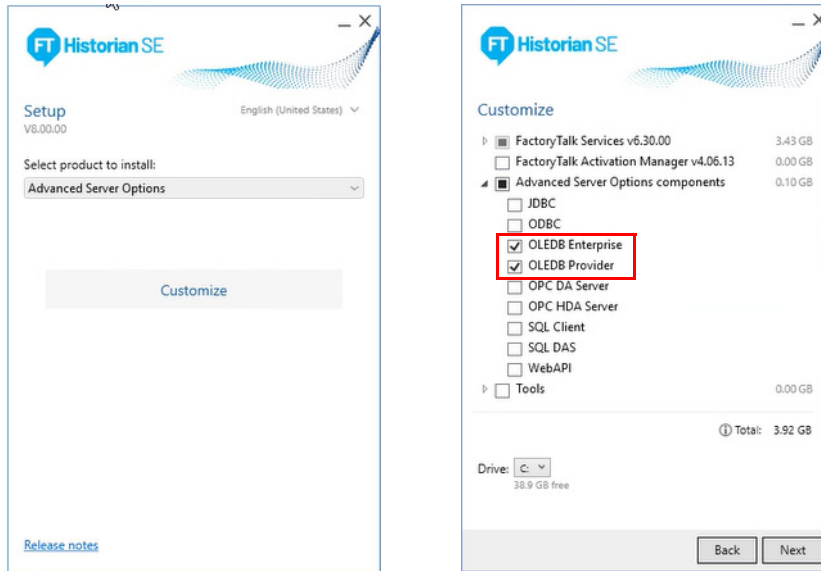
Use the following steps to verify that your system is ready:

- Verify that both OLEDB software components are installed on the SQL Server computer using Windows Programs and Features.

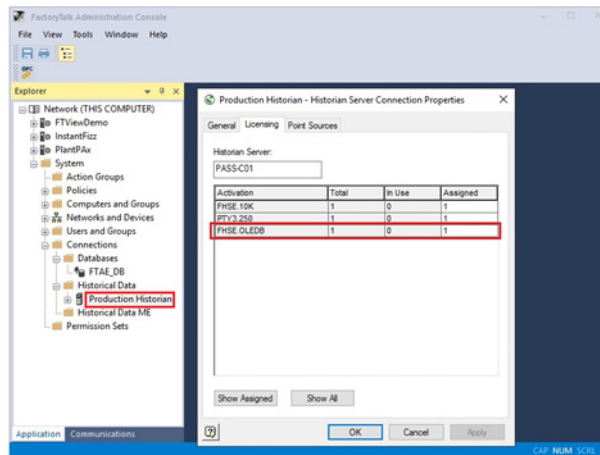


- If the software is not installed, run the Setup.exe file from the FactoryTalk Historian SE software installation media.
- Select “Advanced Server Options” as the product to install and then “Customize”.

4. Select OLEDB Enterprise and OLEDB Provider and then Next.



5. Use the FactoryTalk Activation Manager to verify that the Historian OLEDB Connectivity is activated. A separate activation is required for the OLEDB license.
6. Use the FactoryTalk Administration Console to verify that the Historian OLEDB Connectivity is assigned. Connectivity is configured on the license tab of the historian server connection properties dialog.



Historian Asset Framework Library

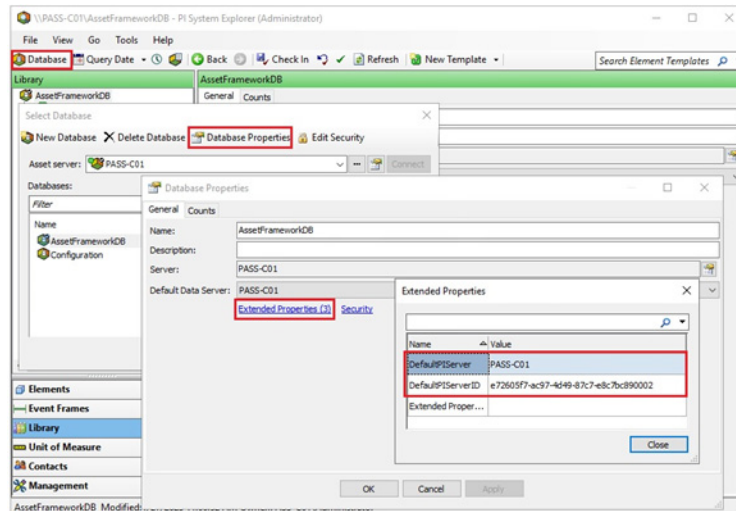
Asset Framework (AF) is provided with FactoryTalk Historian SE to add context to data in the time series historical database.

Using the PI System Explorer:

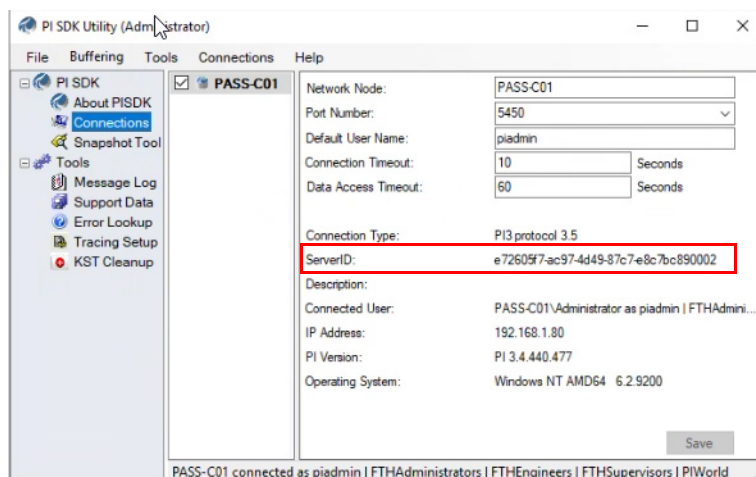
1. Verify that the Asset Framework database has been created.
2. Select 'Database' and record the name of the AF database for use during PlantPax Reporting installation.

IMPORTANT Process Libraries (v5.10.00 or earlier) report templates expect the Asset Framework database to be named "AssetFrameworkDB". Change the database name or use a new Process Library.

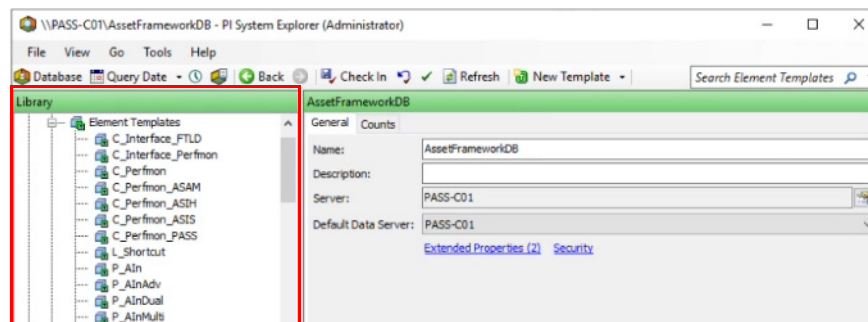
- Click 'Database Properties', click 'Extended Properties' and verify that the AF database is linked to the correct Historian Server ID. If you are unsure about the server ID, you can locate the ID as seen in the following example in the PI SDK utility.



The Historian Server ID can be found on the connection tab of the PI SDK Utility.

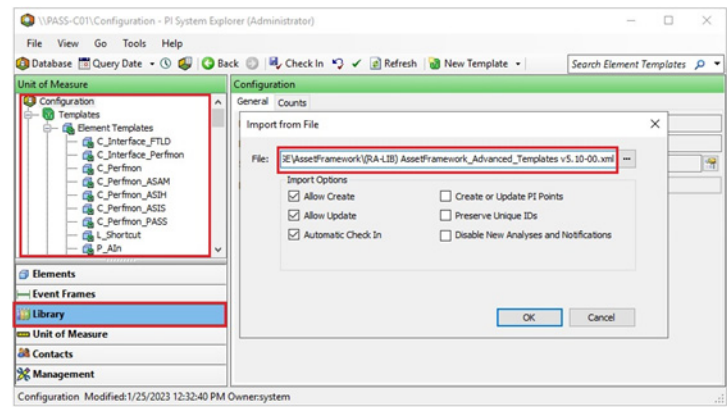


- Next click 'Library' on the lower left.
- Expand Configuration > Templates > Element Templates and verify the PlantPax AF Library Templates have already been imported. If correct, proceed to the next section.



- If unsure whether the correct templates have been imported, re-import them by right-clicking on 'Configuration' on the top left and select 'Import from File'.
- Navigate to Process_Library_V5.xx.xx\Process_Library\Templates\Factory Talk Historian SE\AssetFramework\RA-LIB_AssetFramework_Advanced_Templates v5.xx-xx.xml and open the file.

8. Select the desired Import Options and click OK.



PlantPax EventQ Solution

If the EventQ solution logs to a FactoryTalk Batch SQL database, then the FactoryTalk Batch database must reside on the same reporting SQL server to use the Object Event Report. Otherwise, a separate EventQ database must be created on the reporting SQL server.

ThingWorx Server

PlantPax templates add process-oriented functionality to systems that use PTC ThingWorx software. When using PlantPax templates with a ThingWorx server, the following software and extensions are recommended.

Required Components	Description
Centralized Database	Installed on a Microsoft SQL server computer, with all the components that are required to support the Standard Reports
FactoryTalk Linx Gateway	Installed on a Process Automation System Server (PASS) computer. (Required for Live Data when using Control Strategy Health Dashboards or Vuforia Studio)
Java	OpenJDK or Java 11
SQL Database for ThingWorx	PostgreSQL version 10 or later or Microsoft SQL Server 2014 or later
ThingWorx Foundation	Version 9.2 or later
OPC Aggregator 1	Version 1.9 or later (Required for Live Data when using Control Strategy Health Dashboards or Vuforia Studio)
Vuforia Studio	Version 9.2 or later (Required if using Vuforia)
Vuforia Experience Services	Version 9.2 or later (Required if using Vuforia)
SQL Database for Experience Services	PostgreSQL version 10 or later
Incoming Firewall Rules	Allow ThingWorx HTTP TCP Port 8080 (default) Allow ThingWorx HTTPS TCP Port 8443 (default) Allow Vuforia HTTP TCP Port 3000 (default) Allow Vuforia URL TCP Port 2019 (default) Allow Microsoft SQL Server TCP Port 1433 (default) Allow Microsoft SQL Server UDP Port 1434 (default) Allow PostgreSQL Server TCP Port 5432 (if applicable) Allow OPC UA TCP Port 4990 (default)

PlantPAX Centralized Database

About

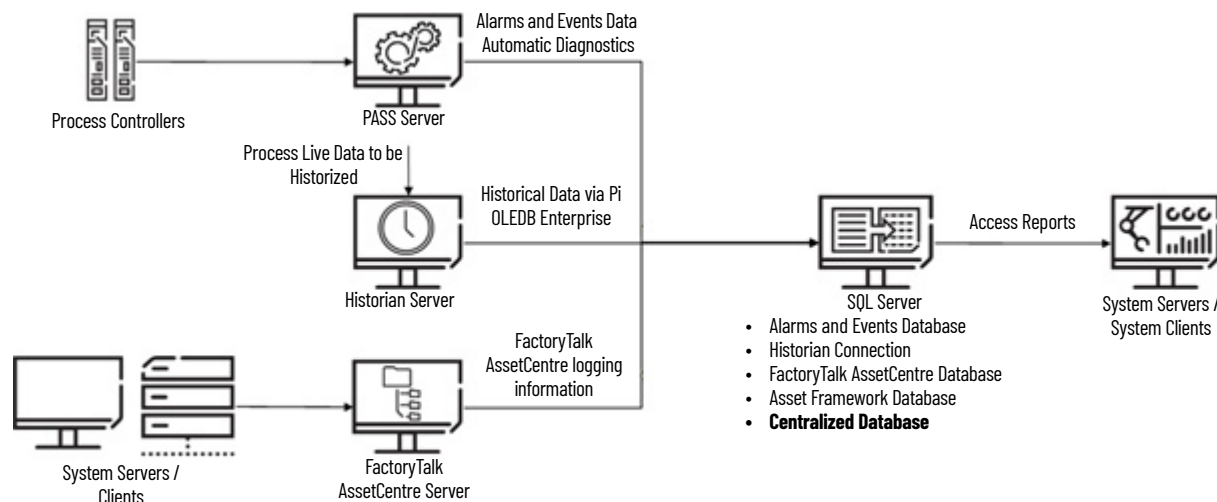
A PlantPax® system contains multiple system databases. The three main databases are FactoryTalk® Alarms & Events, FactoryTalk AssetCentre, and FactoryTalk Historian SE.



Each database structure is unique and works efficiently for the data it's intended to log. The challenge with having different database structures arises when searching for events across them. Different databases often require unique queries and/or analysis tools. Maintenance activities and troubleshooting often require putting ALL the data together to recreate the sequence of events.

PlantPax offers a "Centralized Database" within SQL Server for improved reporting capabilities. The centralized database simplifies plant maintenance. It enables operators and plant managers with a cohesive view of the entire manufacturing floor.

Data Flow Diagram



With the centralized database, data from all applications within your system are consolidated into one location. The data is not changed, only parsed to enable additional features. The consolidation of data allows for improved reporting capabilities with faster loading times.

Reporting platforms, such as SQL Server Reporting Services (SSRS) or ThingWorx®, can query the centralized database for information about Alarms & Events, Process Objects, plus Traceability and System, to give plant-wide maintenance details.

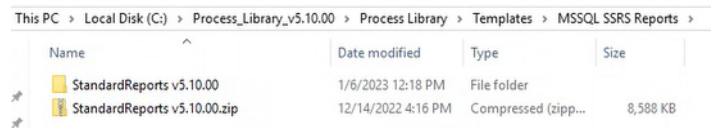
Before the Centralized Database can be created, SSRS must be installed and configured.

Install Standard Reports

The Process Library contains a batch file that is used to install Standard Reports. This script collects user-supplied system information to create the Centralized Database and link it to the system databases within SQL. This script also deploys the reporting webpages for SSRS Reports.

IMPORTANT Before starting this section, confirm that you have collected the information in [Table 1](#)

1. The SQL Server Standard reports are located within the following folder:
"PlantPax_Process_Library_v5.xx.xx\v5.xx.xx\Process Library\Process Library\Templates\MSSQL SSRS Reports"
2. Extract the "StandardReports v5.xx.xx.zip" file.



3. Navigate to the MSSQL SSRS Reports folder within the extracted Zip.
4. Right-click on the file named "StandardReports_5.xx.xx.bat" file and select edit. The file opens in Notepad.
5. In the variables section of the script, change the location for RS.EXE according to the version of the SQL server installed. Comment out the path that is not being used with colons (:). Save the file.

- For SQL Server 2016 and prior, use SET RS="RS.exe"

```

::=====
@echo off
::Variables
::Old location for RS.exe
SET RS="RS.EXE"
::SET RS="C:\Program Files\Microsoft SQL Server Reporting Services\Shared Tools\RS.exe"
::New location for RS.exe

```

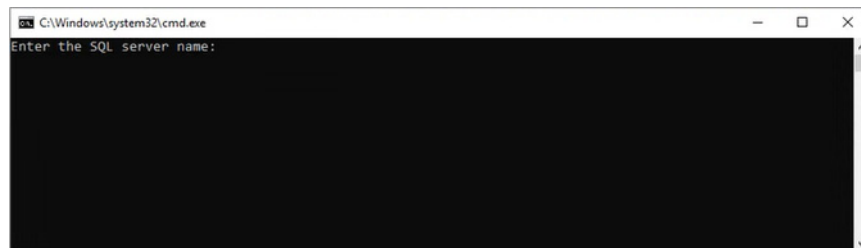
- For SQL Server 2017 and later, use SET RS="C:\Program Files\Microsoft SQL Server Reporting Services\Shared Tools\RS.exe"

```

::=====
@echo off
::Variables
::Old location for RS.exe
::SET RS="RS.EXE"
SET RS="C:\Program Files\Microsoft SQL Server Reporting Services\Shared Tools\RS.exe"
::New location for RS.exe

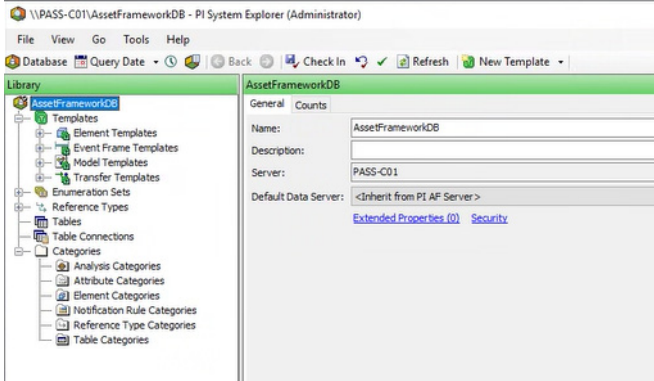
```

6. Save the changes and close Notepad.
7. Right-click the "StandardReports v5.xx.xx.bat" file and select Run as Administrator.
8. A command script window launches. Enter your system-specific parameters as described in [Table 1](#).

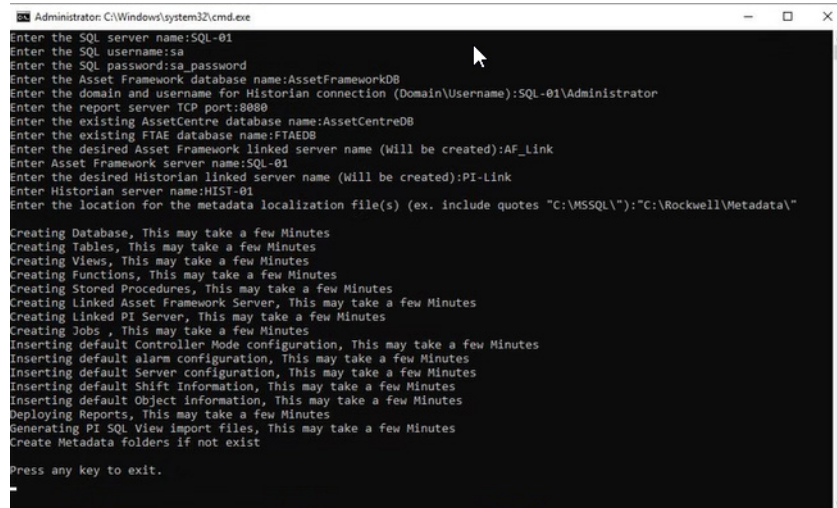


IMPORTANT If an error is made during input, close the script file and start over at [step 7](#). Do not proceed to the completion of the script.

Table 1 - Batch File Parameters

Entry Field	Notes
Enter the SQL server name	<ul style="list-style-type: none"> If using the default instance made during the installation of an SQL server, enter the computer name in this field. If using a Named Instance, enter Server Name\Named Instance in this field. (Example: SQL-01\MyInstance) For computers on a Windows® domain, enter the fully qualified domain name (FQDN). Example: SQL-01.PlantPax.Local
Enter the SQL Username	This user must have sysadmin rights (Example: SA)
Enter the SQL Password	(Example: *****)
Enter the Asset Framework database name	<p>To achieve correct linkage of servers, this name must be the same name as the Asset Framework database name as entered in the PI System Explorer.</p>  <ul style="list-style-type: none"> This becomes the Centralized Database that is used for the reports. (Example AssetFrameworkDB)
Enter the domain and username for the Historian Connection (Domain/Username)	<p>Enter Mapped Historian User (Example: Domain\HistorianUser)</p> <ul style="list-style-type: none"> This Domain\Username is used to authenticate the linked servers. The username cannot contain a space.
Enter the Report server TCP Port	Use the port that was configured in the SQL Reporting Services Configuration section. (Example: 8080)
Enter the existing AssetCentre database name	This is specified on the FactoryTalk AssetCentre Server data source configuration. (Example: AssetCentre_DB)
Enter the existing FTAE database name	This is specified on the FactoryTalk Linx server properties on the Alarms and Events tab. (Example: FTAE_DB)
Enter the desired Asset Framework linked server name (will be created)	This creates a linked server connection to Asset Framework within SQL. (Example: AF_Link)
Enter the Asset Framework server name	<p>This is the computer name where Asset Framework is installed. (Example: SQL-01)</p> <p>For computers on a Windows domain, enter the fully qualified domain name (FQDN). (Example: SQL-01.PlantPax.Local)</p>
Desired Historian linked server name	This creates a linked server connection to Historian within SQL. (Example: PI_Link)
Enter the Historian server name	<p>This is the computer name where Asset Framework is installed. (Example: HIST-01)</p> <p>For computers on a Windows domain, enter the fully qualified domain name (FQDN). (Example: HIST-01.PlantPax.Local)</p>
Enter the working location for the metadata localization files	<p>Specify the location where new or modified metadata localization files are placed prior to inserting them into the centralized database. These files are generated from ACM or the PlantPax Configuration Tool.</p> <p>Include quotes. (Example: "C:\Rockwell\MetaData\" or C:\MSSQL\)</p>

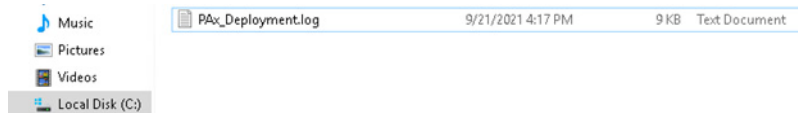
9. After the last parameter is entered, press Enter to run the script.
Review the script step for errors. A successful installation looks similar to the following example.



```

Administrator: C:\Windows\system32\cmd.exe
Enter the SQL server name:SQL-01
Enter the SQL username:sa
Enter the SQL password:sa_password
Enter the Asset Framework database name:AssetFrameworkDB
Enter the domain and username for Historian connection (Domain\Username):SQL-01\Administrator
Enter the report server TCP port:8080
Enter the existing AssetCentre database name:AssetCentreDB
Enter the existing FTA database name:FTADB
Enter the desired Asset Framework linked server name (Will be created):AF_Link
Enter Asset Framework server name:SQL-01
Enter the desired Historian linked server name (Will be created):PI-Link
Enter Historian server name:HIST-01
Enter the location for the metadata localization file(s) (ex. include quotes "C:\MSSQL\");"C:\Rockwell\Metadata\
Creating Database, This may take a few Minutes
Creating Tables, This may take a few Minutes
Creating Views, This may take a few Minutes
Creating Functions, This may take a few Minutes
Creating Stored Procedures, This may take a few Minutes
Creating Linked Asset Framework Server, This may take a few Minutes
Creating Linked PI Server, This may take a few Minutes
Creating Jobs , This may take a few Minutes
Inserting default Controller Mode configuration, This may take a few Minutes
Inserting default alarm configuration, This may take a few Minutes
Inserting default Shift Information, This may take a few Minutes
Inserting default Object Information, This may take a few Minutes
Deploying Reports, This may take a few Minutes
Generating PI SQL View import files, This may take a few Minutes
Create Metadata folders if not exist
Press any key to exit.
  
```

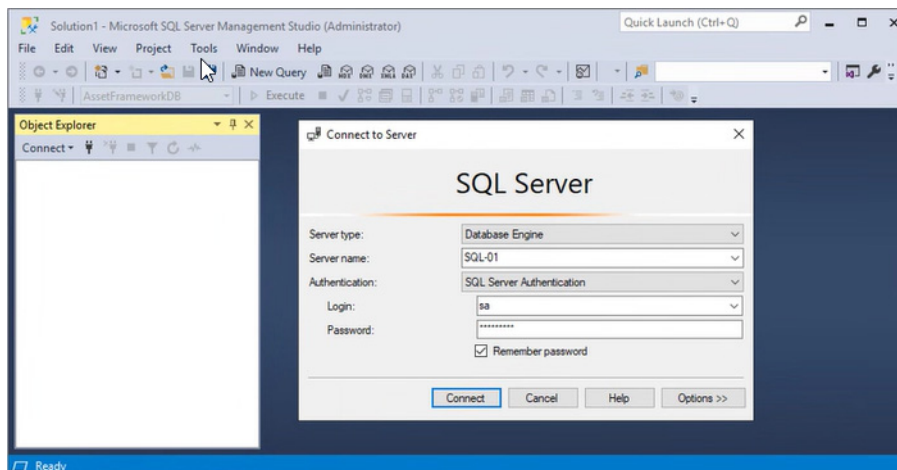
10. To help with troubleshooting errors, a detailed log of the deployment can be found within the root of your C: drive. This captures more detailed information about the deployment of the reports. If there are errors See [Troubleshooting on page 117](#) of this document.



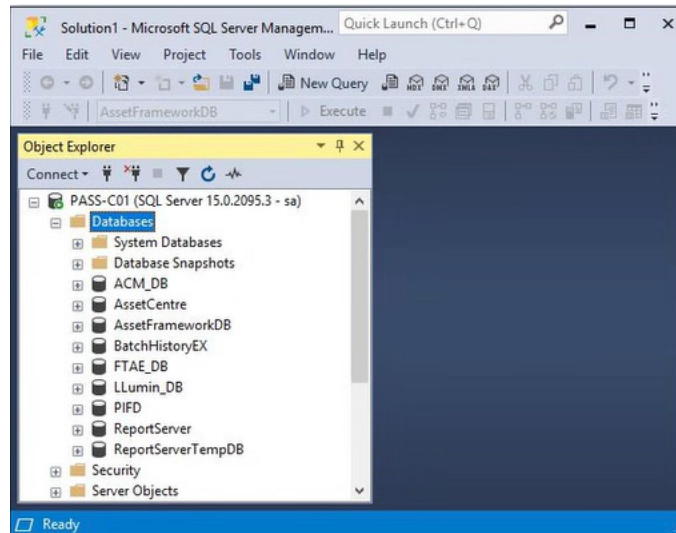
Centralized Database Structure

This section explains the various PlantPax system databases and describes the reporting components within an SQL server.

1. Launch the SQL Management Studio software and connect to your SQL server.

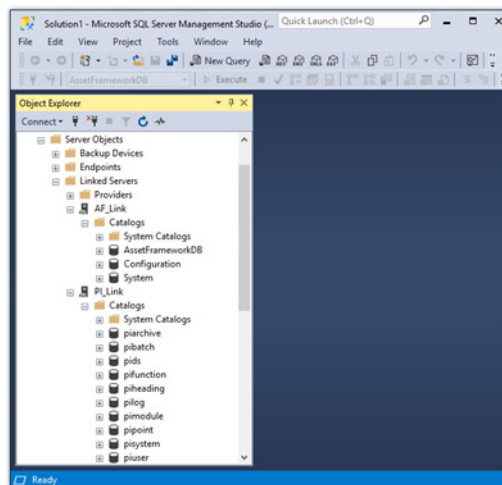


- From the Object Explorer, expand the “Databases” folder.



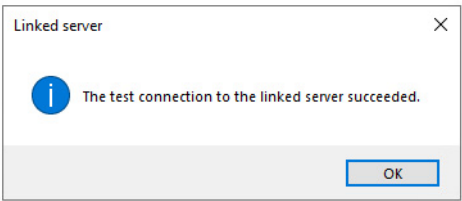
Database	Description
ACM_DB	Application Code Manager project database (if applicable)
AssetCentre	FactoryTalk AssetCentre server database
AssetFrameworkDB	Centralized Database (Created using the same name of your Asset Framework database)
BatchHistoryEx	FactoryTalk Batch database (if applicable) where EventQ Solution data is typically stored.
FTAE_DB	FactoryTalk Alarms and Events database
LLumin_DB	FactoryTalk AssetCentre Management of Change (MoC) database
PIFD	Asset Framework PI foundation database (if applicable)
ReportServer	SSRS primary report database
ReportServerTempDB	SSRS temporary report database

- From the Object Explorer, expand Server Objects > Linked Servers folder. These links were created by the Standard Reports installation script and allow SQL Server to access the Asset Framework and Historian PI databases.

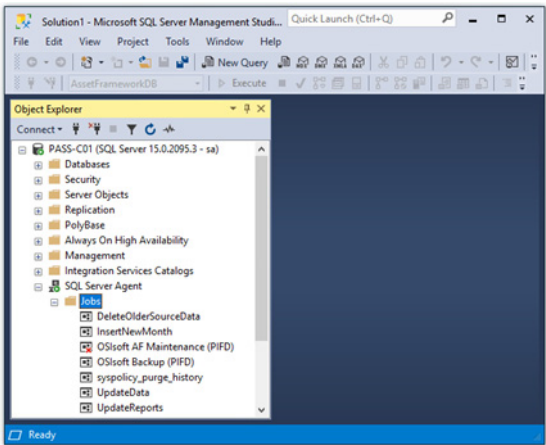


Linked Server	Description
AF_Link	Linked server using data source provider OLEDB Enterprise to connect to the Asset Framework database.
PI_Link	Linked server using data source provider OLE DB Provider to connect to the Historian PI database.


4. Right-click on each “Linked Server” and select “Test Connection”. Verify that the connection test succeeded.



5. If the connection is unsuccessful, record the error message and See [Troubleshooting on page 117](#) of this document.
6. From the Object Explorer, expand the SQL Server Agent > Jobs folder.



Job	Description
DeleteOlderSourceData	PlantPax reporting job removes past data from the centralized database. Manually executed job using a default setting of removing last 3 months of data.
InsertNewMonth	PlantPax reporting job inserts blank data for a new month. Scheduled to run on the 25th of each month. Useful for viewing consecutive days within reports, such as when control equipment might have been offline.
UpdateData	PlantPax reporting job reads new data from the source databases and places it into the centralized database. Scheduled to run every 15 minutes.
UpdateReports	PlantPax reporting job processes data within the centralized database. It calculates the number of stops, starts, faults, and so on, from every event frame and creates historical shift reports. Scheduled to run every 15 minutes.

 Depending upon the performance of the SQL server computer, there may be overlap where the jobs UpdateData and UpdateReports take longer than 15 minutes to complete. Edit the jobs and extend the schedule to run every once an hour.

Controller Localization Files

PlantPax reporting uses controller tag metadata to provide extended data such as descriptions, engineering units, object class, and other information for the report webpages.

Controller localization text files are required for new reporting installations and whenever new tags or control strategies are added to an existing controller application.

Generate Localization Files

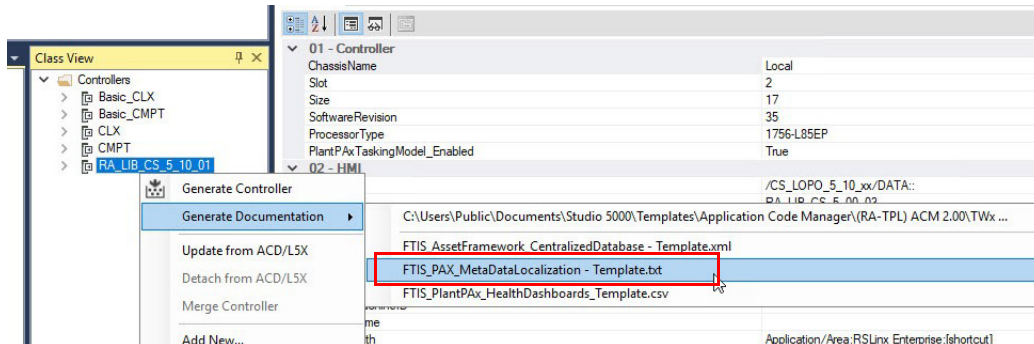
There are two software tools for generating tag metadata to Logix Localization files.

- Application Code Manager (ACM)
- PlantPax Configuration Tool

Application Code Manager (Option 1)

ACM software is the preferred method for developing PlantPax applications and generating application content. If the controller application was created with ACM, use this method.

1. Open the ACM project.
2. In the Classic View window, right-click on the controller and select 'Generate Documentation'.



3. Select the MetadataLocalization – Template.txt option.
4. Place the files in a directory on the SQL server computer that will be used for reporting. (Ex: C:\Rockwell\Metadata\ or C:\MSSQL\)

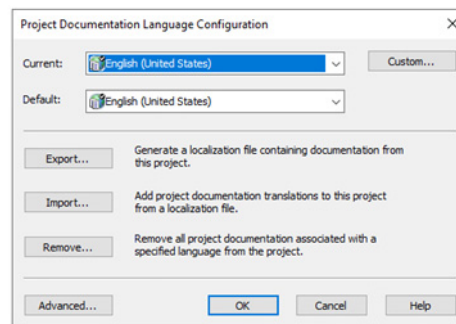


Confirm you have full access to the metadata directory. A read-only attribute results in an access-denied message from the SQL stored procedure extended data import.

PlantPax Configuration Tool (Option 2)

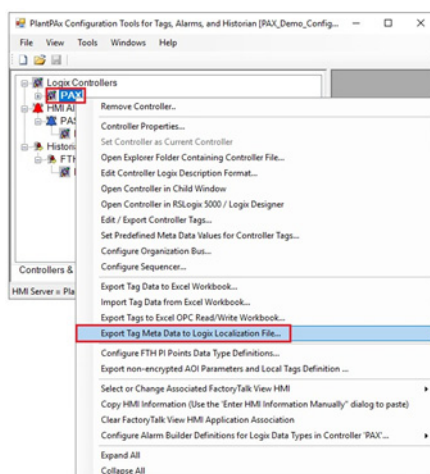
The PlantPax Configuration tool is provided with the Process Library download. You can install it and use it on an engineering workstation and use it for various system functions. This tool is intended for controller applications that have been developed without ACM or later modified outside of ACM.

1. Before opening your controller application file (.ACD) with the PlantPax Configuration Tool, confirm that at least one localized language has been assigned.
2. To verify you have a localized language assigned, open Studio 5000 Logix Designer®, and open your application file (.ACD).
3. On the top menu, select the Tools > Documentation Languages
4. Use the drop downs to select your desired language support.



5. Click OK and save your Studio 5000 Logix Designer project.
6. Launch the PlantPax Configuration Tool.
7. Open an existing project or create a project with your controllers, HMI, and Historian information.
8. Expand the Logix Controllers category and right-click on a controller.

- Select the option 'Export Tag Meta Data to Logix Localization File'.



- Place the files in a directory on the SQL server computer that will be used for reporting. (Ex: C:\Rockwell\Metadata\ or C:\MSSQL\)



Confirm you have full access to the metadata directory. A read-only attribute results in an access-denied message from the SQL stored procedure extended data import.

Localization File Modification

Both ACM and PlantPax configuration create the metadata localization files with an internal source name of the Logix Designer application ACD file name. The PlantPax centralized database requires that the source name be changed to the FactoryTalk Linx shortcut name for the given controller.

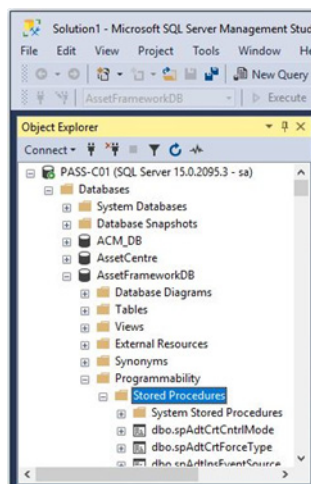
- Browse the directory on the SQL server computer with the metadata files. (Ex: C:\Rockwell\Metadata\ or C:\MSSQL\)
- Open each controller metadata file using Notepad and change the name in the second line labeled #Source: "ShortcutName"
- Do not add brackets or quotes. Save the file and proceed to the Import sections.

Initial Extended Data Import

New centralized database installations require a one-time manual execution of a SQL stored procedure that is used for initial update of Metadata Extended Tag Properties and Asset Framework Snapshot into the centralized database.

- Launch the SQL Management Studio software (SSMS) and connect to your SQL server.
- From the Object Explorer, navigate to Databases > (Centralized Database) > Programmability > Stored Procedures

Where (Centralized Database) is the name that you used for your Asset Framework Database. In the following example, AssetFrameworkDB is used.



- Browse the Stored Procedures folder for a procedure named "dbo.spComUpdInitial".

4. Right-click on "dbo.spComUpdInitial" and select "Execute Stored Procedure".
5. Wait for the query to complete before closing SSMS. This may take several hours depending on the size of the exiting databases.
6. Verify if the data was imported to the object dictionary. Select the "New Query" button and type the following into the query window. Then select "Execute".

```
select *
from vObjectDictionary
where TagElement is null
AND ExtendedType = 'Description'
```

7. Verify that the object class is not "null" for your main process objects. The following example shows query results for a PMTR object that is named MT1101 that has an object class of Motor and a PAI object that is named XT520 has an object class of Analog. The other tags or members that are listed as "null" are not process objects.

	ControllerTopic	ProgramID	Program	TagID	Tag	ObjectClassID	ObjectClass	ClassVar
224	[PAX]	1	eTK101_Example	755	XV1101_intlk_5	NULL	NULL	NULL
225	[PAX]	1	eTK101_Example	275	MT1101	3	Motor	58
226	[PAX]	NULL	NULL	224	HILO590	NULL	NULL	NULL
227	[PAX]	NULL	NULL	97	XT520	1	Analog	48
228	[PAX]	NULL	NULL	561	TK1102_EM_RPT_04	NULL	NULL	NULL
229	[PAX]	NULL	NULL	677	XIC560_intlk_5	NULL	NULL	NULL
230	[PAX]	NULL	NULL	265	MT101_intlk_0	NULL	NULL	NULL

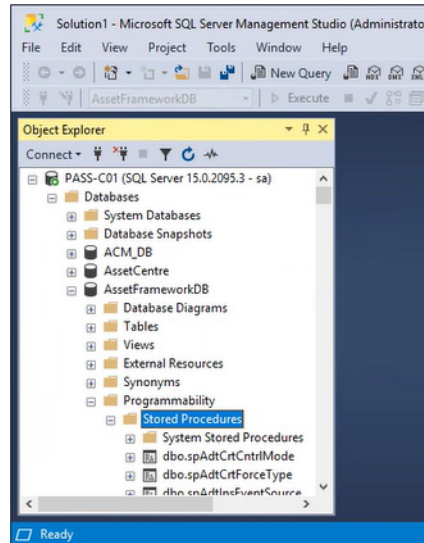
8. Browse your metadata file directory. It now contains a "Completed" subdirectory that contains time stamp localization files that were processed into the centralized database.

Extended Data Import

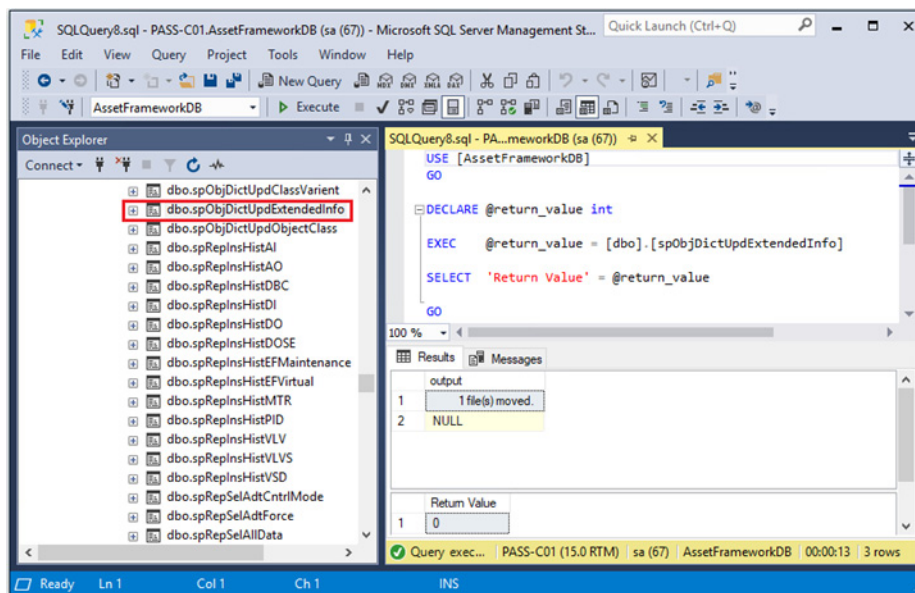
This procedure inserts modified and new controller extended tag properties from the metadata localization files into the centralized database. Perform this whenever an additional controller or control strategy has been added.

1. On the SQL server computer, use the Windows file explorer navigate to the directory that you specified for metadata localization files and verify that your files are present. (ex: C:\Rockwell\Metadata or C:\MSSQL\)
2. The directory will contain a "Completed" subdirectory that was created after running the initial extended data stored procedure. For new controller files or modified existing controller files, copy them into the root level. (ex: C:\Rockwell\Metadata or C:\MSSQL\)
3. Launch the SQL Management Studio software and connect to your SQL server.
4. From the Object Explorer, navigate to Databases > (*Centralized Database*) > Programmability > Stored Procedures

Where (*Centralized Database*) is the name that you used for your Asset Framework Database. In the following example, AssetFrameworkDB is used.



5. Scroll through the list of stored procedures until you locate the one named "dbo.spObjDictUpdExtendedInfo".
6. Right-click on "dbo.spObjDictUpdExtendedInfo" and select "Execute Stored Procedure".



7. Return to the Windows file explorer that is viewing the localization files and navigate into the sub directory named "Completed."
8. Verify that the completed folder contains your time-stamped metadata localization files that were processed by the stored procedure.
9. If the stored procedure execution was unsuccessful, record the error message and See [Troubleshooting on page 117](#) of this document.

Historian Asset Framework Elements

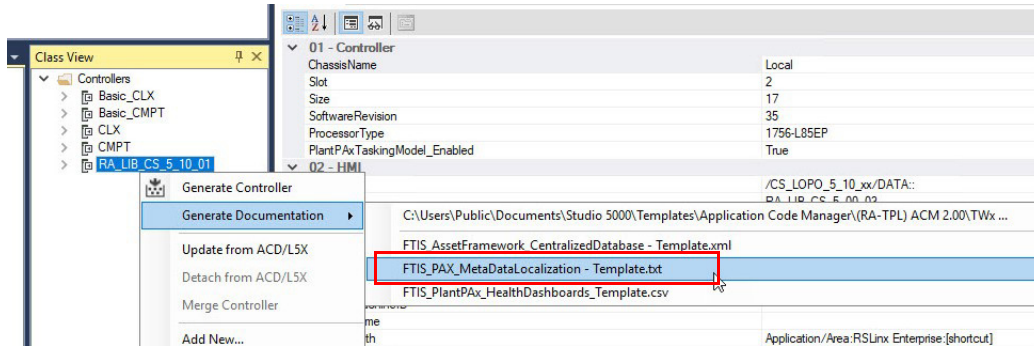
Asset Framework elements provide a means to organize your process equipment assets. For PlantPax system purposes, “element” can be considered synonymous with “objects” in the Rockwell Automation Library of Process Objects.

There are two software tools for creating Asset Framework Elements:

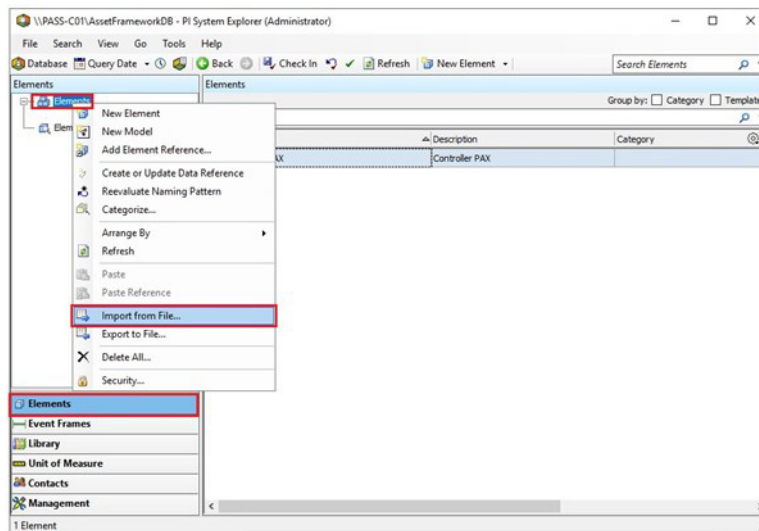
- Application Code Manager (ACM)
- PlantPax Configuration Tool

Application Code Manager (Option 1)

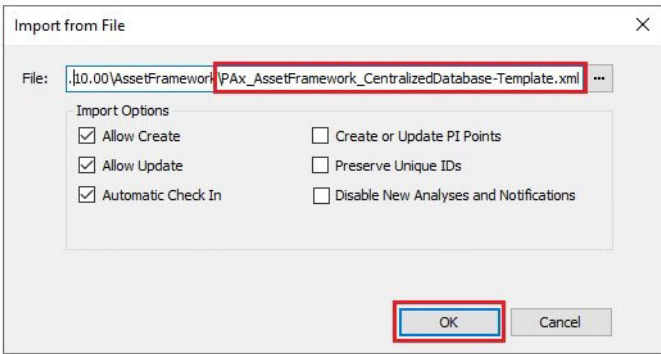
1. Open the ACM project.
2. In the Classic View window, right-click on the controller and select Generate Documentation.



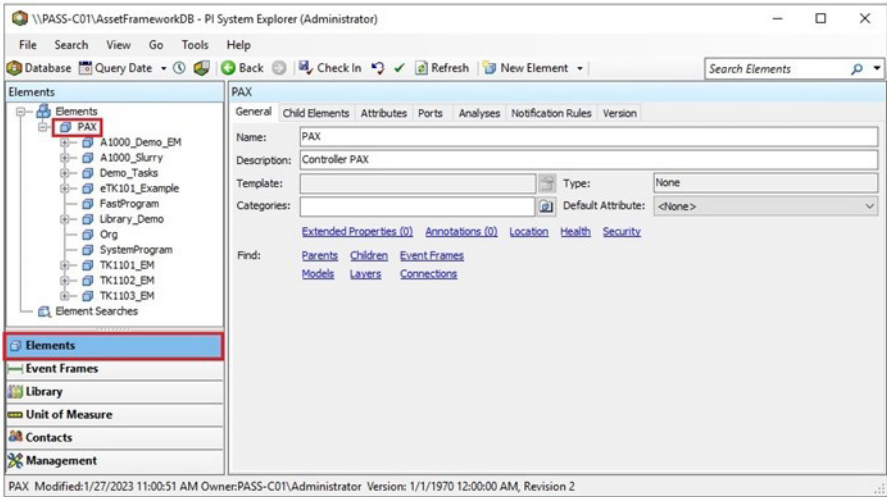
3. Select the AssetFramework.CentralizedDatabase- Template.txt option.
4. Place the files to a temporary directory on the Asset Framework computer.
5. Using a PI Client or the Asset Framework computer, launch the PI System Explorer.
6. Select “Elements” on the lower left.
7. Right-click on Elements in the left window and select “Import from File”.



8. Browse to the file location where the element template created with ACM is located.



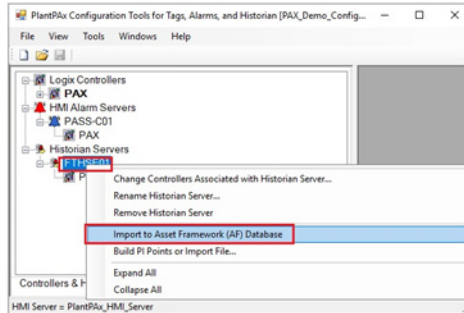
9. After the import completes, expand "Elements" and verify that the controller project elements exist.



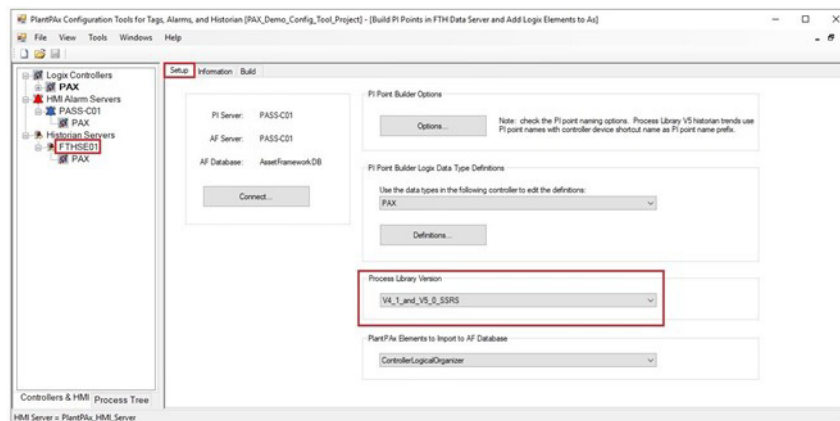
PlantPax Configuration Tool (Option 2)

The PlantPax Configuration tool is provided with the Process Library download. You can install it and use it for various functions. This tool is intended for controller applications that have been developed without ACM or later modified outside of ACM.

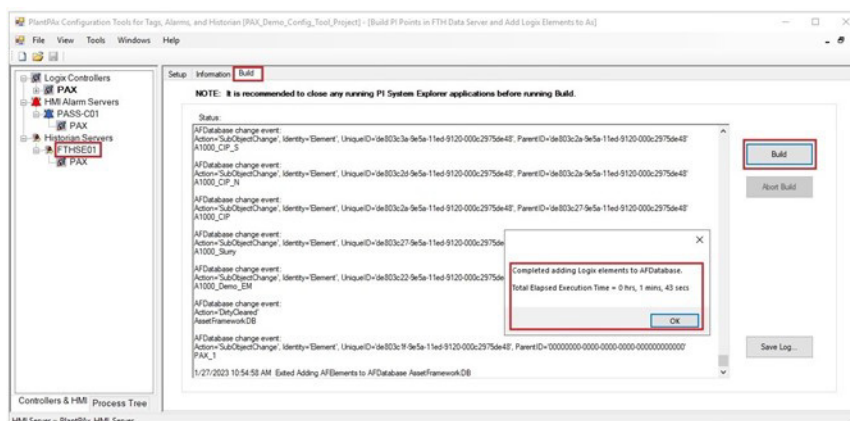
1. Launch the PlantPax Configuration Tool.
2. Open an existing project or create a project with your controllers, HMI, and Historian information.
3. Select the option "Import to Asset Framework (AF) Database".



4. Modify both the PI Point Builder Options and then the Process Library Version to V4_1_and_V5_0_SSRS.

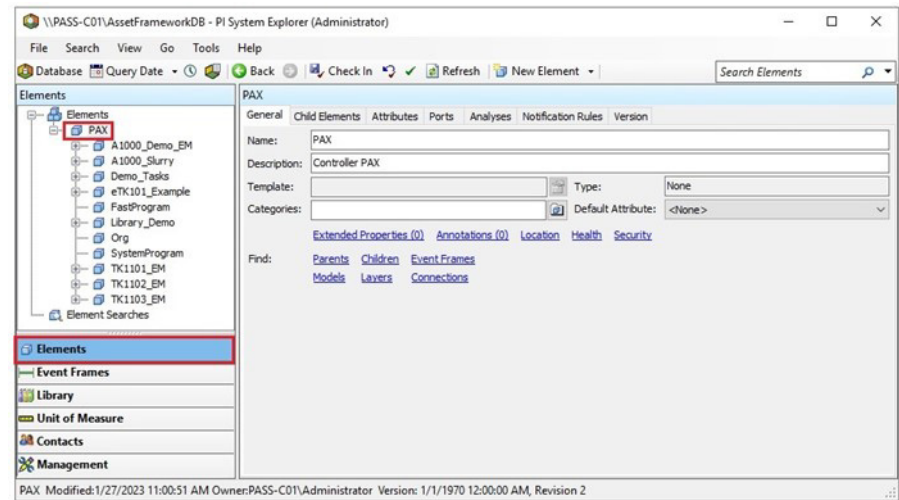


5. Select the "Build" tab and the click the "Build" button. Build adds the Logix elements directly into the AssetFramework database without the need for a separate import.



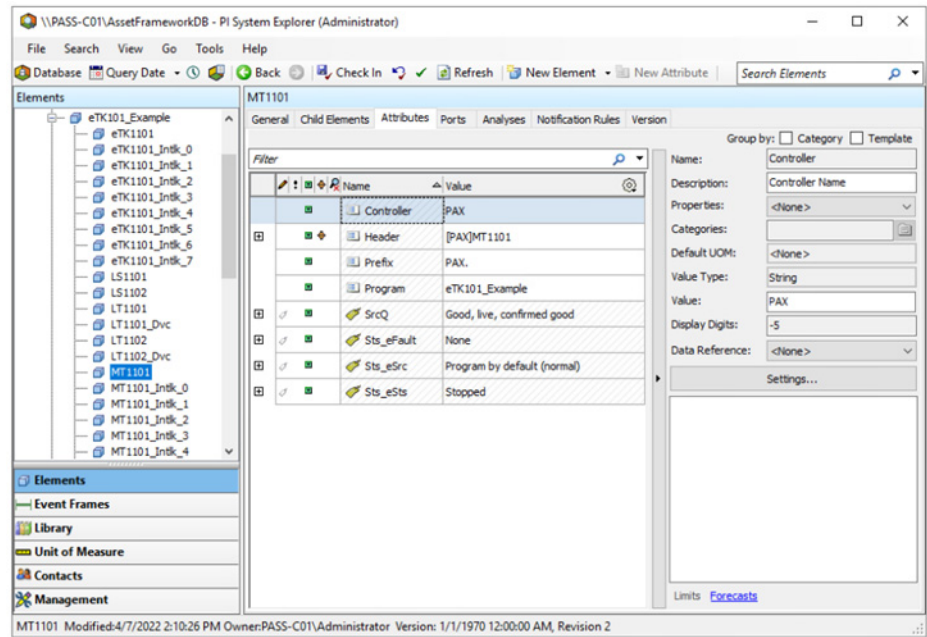
6. Select OK when finished.
7. Using a PI Client or the Asset Framework computer, launch the PI System Explorer.
8. Select "Elements" on the lower left.

9. Expand “Elements” and verify that the controller project elements exist.



Verify Historian Connection

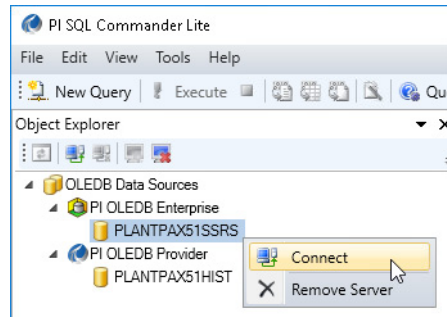
- 1. Expand the Elements for a given controller and check a few process objects for good connection to the Historian database.
- 2. Select the Attributes tab and verify that there are no “bad input” messages for values.



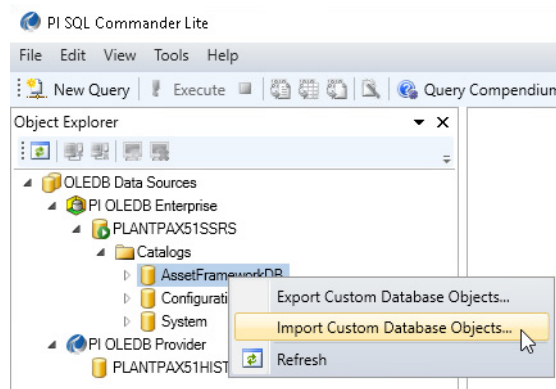
Pi SQL Commander Lite Data Views

Pi SQL Commander Lite is an application that you must use to navigate a relational view of our PI system. This section describes how to import a template created by the Standard Report script. The template creates data views, which the centralized database uses to access the Archive and Snapshot data, including errors.

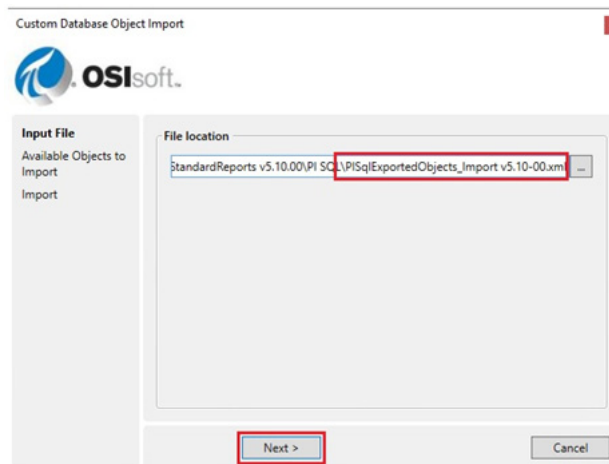
1. On the SQL server, navigate to Pi System > Pi SQL Commander Lite(64-bit) and open the application.
2. Connect to the Pi OLEDB Enterprise by right-clicking on the SQL server name.



3. Browse to your AssetFramework DB in the Pi Commander window via Catalogs -> "AssetFramework DB Name"
4. Right-click on your AssetFramework DB and select "Import Custom Database Objects".

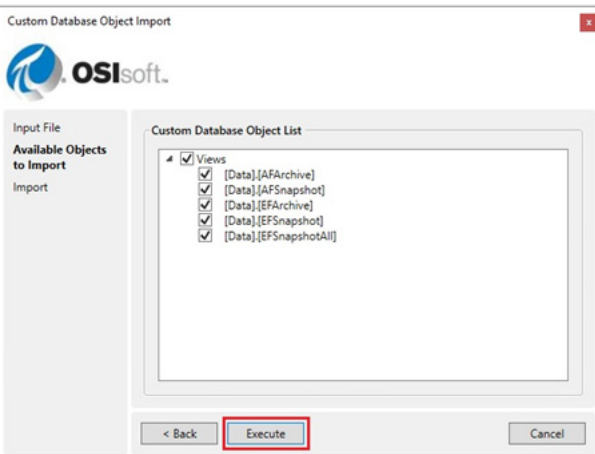


5. Select the file ellipses and Navigate to the following folder "Process_Library_v5.xx.xx\Process Library\Templates\MSSQL SSRS Reports\StandardReports v5.xx.xx\PI SQL"
6. There are two files in the PI SQL directory. Select the file containing the word "Import". (Ex: "PISqlExportedObjects_Import v5.xx-xx.xml").



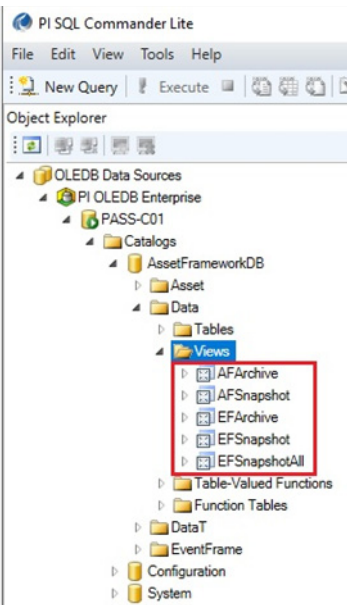
7. Select Next

8. Select Execute



9. Select Done

When Complete there should be five views under Data > Views



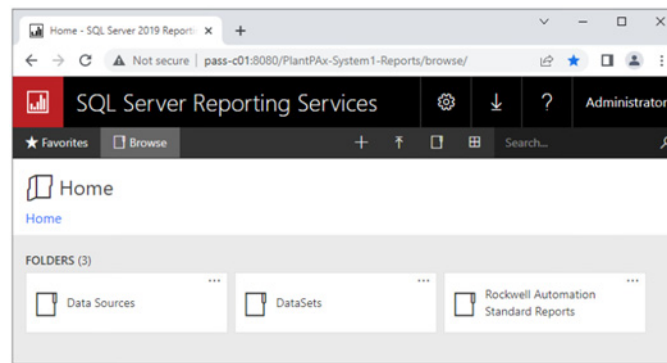
Rockwell Automation Standard Reports

Rockwell Automation Standard Reports is the base reporting option for a PlantPAx® 5.0 system. SQL Server Reporting Services (SSRS) hosts reporting webpages that display content from the centralized database.

About

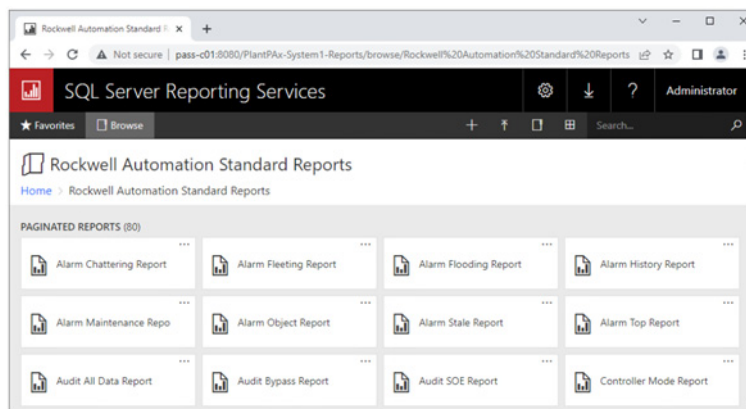
Reports webpages present Alarms & Events, Process Objects, and Traceability & System information to display historical system data in a meaningful layout. This means PlantPAx operators and maintenance engineers can quickly and easily gain insight into plant operations. Reports are viewed either from an internet browser or an embedded link within a FactoryTalk® View SE HMI application.

1. To start viewing the standard reports, open your web browser and type in the web portal URL that was configured within SSRS. For example, [http://\[ComputerName\]:8080/PlantPAx-System1-Reports](http://[ComputerName]:8080/PlantPAx-System1-Reports).



Folder	Description
Data Sources	connection details to the Centralized Database
Data Sets	contains information to retrieve specific sets of data from data source
Rockwell Automation Standard Reports	PlantPAx report webpages

2. Select the "Rockwell Automation Standard Reports" button.



Report types are listed alphabetically and grouped as follows:

Report Type	Description
Alarm	Provides insight for Chattering, Fleeting, Flooding, History, Maintenance, Object, Stale, and Top.
Audit	Provides traceability for All, Bypass, and Sequence of Events.
Object	Provides reports for the Process Controller process objects for a given Month, Day, and Shift.
System	Provides Audit, Force, Maintenance, Metadata, and Virtual information.

Report Navigation

Alarm Reports

Alarm reports are derived using data from the FactoryTalk Alarms and Events database. Each alarm report is processed using stored procedures within the centralized database. These reports are intended to help End Users comply with the alarm management lifecycle that is documented in the ANSI/ISA-18.2-2016 standard and similar guidelines and recommendations.

Alarm Report	Description
Chattering Alarm Report	Alarm chattering is diagnosed when there are 3 or more alarm activations in one minute. Thereafter, alarm chattering is considered to have ended if there are no more alarm activations within 60 seconds of last activation.
Fleeting Alarm Report	A fleeting alarm is an alarm that loses its active status within 10 seconds of becoming active.
Alarm Flooding Report	The reporting interval is subdivided into a series of 10-minute intervals. Consecutive intervals is a period. A flooded period is initiated by a flood interval with an alarm count > 10 and is concluded by a recovery interval with an alarm count < 5. All intermediate intervals are also considered as flooded intervals. The recovery interval is not a flooded interval and is not part of the flooded period. Alarm-flooding is based on flooded intervals/period from the perspective of each group path.
Alarm History Report	Alarm history report for all areas. Report can be filtered by time/date, per user, alarm priority, alarm type, alarm attribute, alarm state and area.
Alarm Maintenance Report	Graphical compilation of chattering, fleeting, stale, and flooding alarms.
Stale Alarm Report	Alarms that are active for longer than 24 hours are classified as stale alarms.
Alarm Top Report	Top alarm report for all areas. Report can be filtered by area and alarm priority.

Audit Reports

Audit reports are derived using data from the FactoryTalk Alarms and Events, and the FactoryTalk AssetCentre database.

Audit Report	Description
Audit All Data Report	Audit All Data reports are filtered by time/date, object class and tag name. Object classes include Analog, Digital, Motor, Valve, Procedural Control, Regulatory Control, Cross-Functional, Built in Instructions, Controller Diagnostics, Computer Perfmon, Compressor, Material Handling, Device Check and System.
Audit Bypass Report	Audit Bypass reports are filtered by time/date and display the tag, computer, and person who bypassed an object, such as a control strategy permissive.
Audit SOE Report	Audit SOE reports are filtered by time/date and display the sequence of events of actions that occurred within the FactoryTalk system. Event sources include FactoryTalk Activation, FactoryTalk Security, FactoryTalk Alarms and Events, FactoryTalk Logix Echo Service, FactoryTalk Linx Gateway DA, FactoryTalk Linx OPCServer, Logix Designer application, and so on.
Controller Mode Report	Controller Mode reports are filtered by time/date and display any Logix Controller Mode changes to Program or Run.
System Force Report	System Force reports are filtered by time/date and display any Logix Controllers Forces by Type, Person, Controller, and Message.
System Maintenance Report	System maintenance reports are filtered by time/date and display objects that are/were in maintenance with a duration of the maintenance period.
System Metadata Report	System metadata reports are filtered by time/date and display objects that had their metadata information changed with details of who changed the information
System Virtual Report	System virtual reports are filtered by time/date and display objects that are/were in virtual mode with a duration of the virtual period.

Object Reports

Object reports are designed for displaying the status of PlantPAx 5.0 process objects. Month, Day, and Shift reports are available for each process object listed. Most object reports allow filtering by Controller topic, Tag scope, Program name, and tag name.

Object Report	Description
Object Audit Report	Object audit reports are filtered by time/date, object class and tag name. Object classes include Analog, Digital, Motor, Valve, Procedural Control, Regulatory Control, Cross-Functional, Built in Instructions, Controller Diagnostics, Computer Perfmon, Compressor, Material Handling, Device Check, and System.
Object EM State Steps Report ⁽¹⁾	Graphical representation of Generic Equipment Module (EM) states and steps over time. Filtered on a start and end time.
Object EM State Report	Graphical representation of steps over time within a given Generic Equipment Module (EM) state. Filtered on a start and end time.
Object EMGen Report ⁽¹⁾	Generic Equipment Module displays states, steps, faults, and availability during specific periods.
Object EP State Step Report ⁽¹⁾	Graphical representation of Generic Equipment Phase (EP) states and steps over time. Filtered on a start and end time.
Object EP State Report ⁽¹⁾	Graphical representation of steps over time within a given Generic Equipment Phase (EP) state. Filtered on a start and end time.
Object EPGen Report ⁽¹⁾	Generic Equipment Phase displays states, steps, faults, and availability during specific periods.
Object Event Report	Provides details on events relating to specific object.

Object Report	Description
Object PAI Report	Process Analog Input displays minimum, maximum, average, and availability during specific periods.
Object PAO Report	Process Analog Output displays minimum, maximum, average, faults, time at target state, and availability during specific periods.
Object PDBC Report	Process deadband Controller displays number of times input went below raise setpoint, number of times input went above lower setpoint, faults, time outside deadband, and availability during specific periods.
Object PDI Report	Process Digital Input displays number of activations, active time, and availability during specific periods
Object PDO Report	Process Digital Output displays the number of activations, faults, active time, and availability during specific periods.
Object PDOSE Report	Process Dosing displays hourly total, daily total, monthly total, and availability during specific periods.
Object PLCPU Report ⁽¹⁾	Process Controller displays average and maximum utilization for each of the 4 cores, as well as minor faults during specific periods.
Object PMTR Report	Process Motor displays the number of starts for each of the two starts, faults, running time, and availability during specific periods.
Object PPID Report	Process Proportional + Integral + Derivative displays minimum, maximum, average, manual control time, time outside deadband, and availability during specific periods.
Object PVLV Report	Process Valve displays the number of times each of the two positions were reached, faults, time at position two, and availability during specific periods.
Object PVLVS Report	Process Valve Statics reports display stroke times and stroke counts.
Object PVSD Report	Process Variable Speed Drive displays the number of starts for each of the two directions, faults, running forward time, and availability during specific periods.
Object SEQ Report ⁽¹⁾	Process Sequencer displays states, steps, faults, and availability during specific periods.
Object SEQ State Step Report ⁽¹⁾	Graphical representation of SEQ states and steps over time. Filtered on a start and end time.
Object SEQ State Report ⁽¹⁾	Graphical representation of steps over time within a given SEQ state. Filtered on a start and end time.
Object Unit Report	Process Unit displays states, faults, and availability during specific periods.
Object Unit State Report ⁽¹⁾	Graphical representation of Unit states over time. Filtered on a start and end time.

(1) Report available with Process Library 5.10.00 or later.

System Reports

System Report	Description
Database Stat Report	Database statistics report for the centralized database displays the size in MB, lists the tables, total rows in each table, and last table update time/date.
Shift Setup Report	Displays the shift configuration. Shift name with start and end time.
System Audit Report	System audit reports display a list of system wide activities. The system audit reports are filtered by time/date for tag name, object class, person, event source and message.
Object Top Logger Basic	Provide details on the number of logs that exist per object based on the originating source. Sources that are shown in the basic report are Alarming (FTAE) and State Changes (Event Frames). Filtered on a start and end time.
Object Top Logger Detailed	Provide details on the number of logs that exist per object based on originating source. Sources that are shown in the Advanced report are Alarming (FTAE), Asset Framework (Asset Framework), and State Changes (Event Frames). Filtered on a start and end time.

Printing Reports

All Rockwell Automation Standard Reports can be saved to multiple formats for sharing or archiving data. These reports are designed to be printed or shared and are formatted to fit well on a page. Simply select the Save icon in the menu and choose your desired format.

Home > Rockwell Automation Standard Reports > Alarm History Report

Start Date: 11/3/2022 6:19:03 AM End Date: 11/3/2022 7:19:03 PM User/Controller Selection: User

Alarm Priority: Select All Alarm Attribute: Select All Alarm State: Select All Area: Select All

Alarm Type: Select All

1 of 2 ? 100%

Word Excel PowerPoint PDF TIFF file MHTML (web archive) CSV (comma delimited) XML file with report data Data Feed

03 Nov 2022 06:19:03

Date	Tag Name	Alarm State	Alarm Description
03 Nov 2022 12:34:15	eTK1101	Active	eTK1101 - TK1101 Equipment Module: Interlock Trip
03 Nov 2022 12:34:15	eTK1101	Active	eTK1101 - TK1101 Equipment Module: Interlock Trip
03 Nov 2022 12:34:15	eTK1101	Active	eTK1101 - TK1101 Equipment Module: Interlock Trip
03 Nov 2022 12:34:15	eTK1101	Active	eTK1101 - TK1101 Equipment Module: Interlock Trip
03 Nov 2022 12:34:15	eTK1101	Active	eTK1101 - TK1101 Equipment Module: Interlock Trip
03 Nov 2022 12:34:15	eTK1101	Active	eTK1101 - TK1101 Equipment Module: Interlock Trip
03 Nov 2022 12:34:15	eTK1101	Active	eTK1101 - TK1101 Equipment Module: Interlock Trip
03 Nov 2022 12:34:15	eTK1101	Active	eTK1101 - TK1101 Equipment Module: Interlock Trip
03 Nov 2022 12:34:15	eTK1101	Active	eTK1101 - TK1101 Equipment Module: Interlock Trip
03 Nov 2022 12:34:15	eTK1101	Active	eTK1101 - TK1101 Equipment Module: Interlock Trip
03 Nov 2022 12:34:15	eTK1101	Active	eTK1101 - TK1101 Equipment Module: Interlock Trip

Interlock Trip

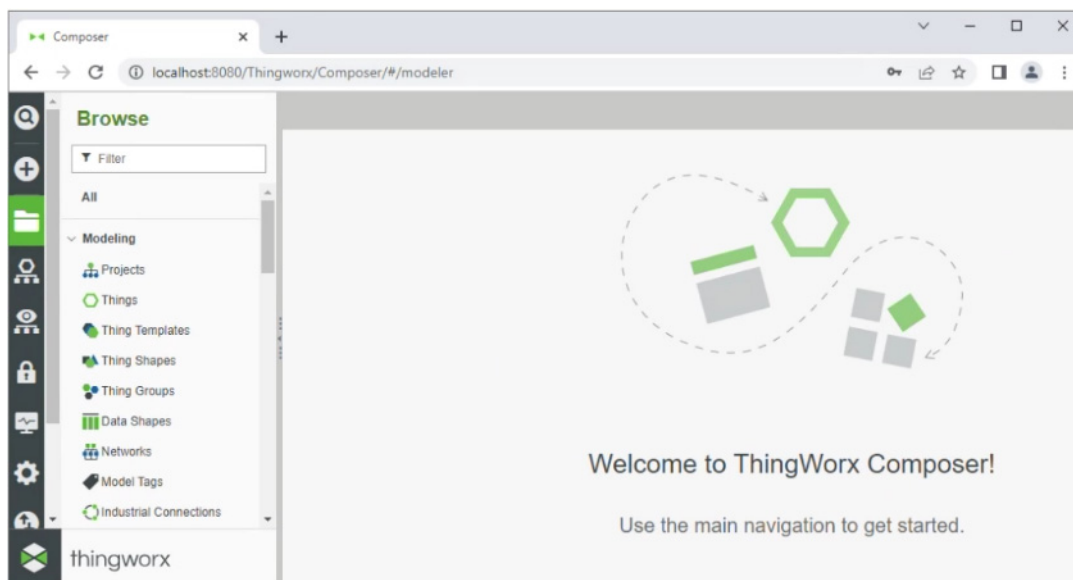
PTC ThingWorx Foundation

The Rockwell Automation Process Library contains maintenance report and control strategy health dashboard templates for a ThingWorx® Foundation server.

About

The ThingWorx Foundation server consumes historical data for reporting from the PlantPAx SQL server and polls live data from Process Controllers.

If you already have a ThingWorx Foundation server, proceed to the [Enable Extension Import](#) section.



Installation

Refer to PTC® for complete installation and upgrade documentation. This section provides basic installation guidance to confirm the PlantPAx® templates will deploy and operate properly.

The ThingWorx Foundation Installer has two main prerequisites before installation:

- Install SQL and create a ThingWorx database with an administrator user
- Install Java and create Windows® environment variables

SQL Database

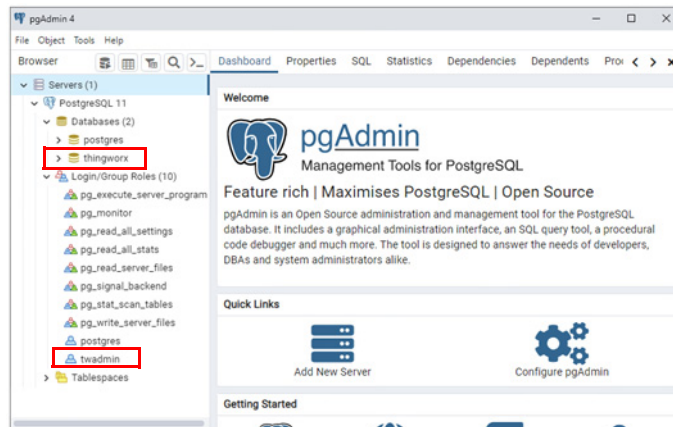
ThingWorx Foundation supports either a Microsoft® SQL Server or PostgreSQL Server (recommended) for the ThingWorx database.

This SQL database is used by ThingWorx and typically separate from the centralized database SQL server. Full featured ThingWorx servers can write a large amount of content to its database, thus a designated database is recommended. For this document, a PostgreSQL Server is installed on the ThingWorx Foundation server computer. The PlantPAx Maintenance Reports read from the centralized database and don't produce large amounts of data locally. Using a PostgreSQL server allows for shared usage with Vuforia Experience Services.

1. Download and install PostgreSQL Server.

PostgreSQL is an open source object-relational database system.

- Launch the pgAdmin management tools for PostgreSQL. pgAdmin is a management tool for your databases that is included in the PostgreSQL installation. This tool is similar to the SQL Management Studio software and allows you to create a ThingWorx database and ThingWorx user.

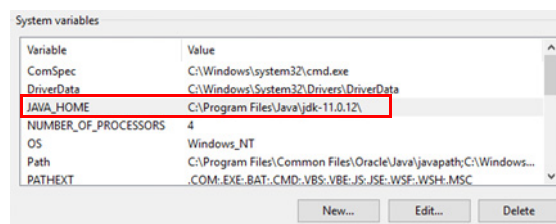


- Right-click the Login/Group Roles folder and create a user named "twadmin" with all privileges enabled for; login, superuser, create roles, create databases, inherit rights from parent roles, initiate streaming replication and backups.
- Right-click the Databases folder and create a database that is named 'thingworx' with the owner of "twadmin".
- Record the "thingworx" database name and "twadmin" username, as they are required for the ThingWorx Foundation Installer.

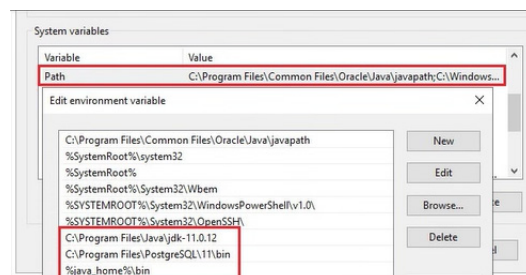
Java SE Development Kit

ThingWorx Foundation uses the Apache Tomcat 9.0 web server. A prerequisite for Tomcat is Java Standard Edition Runtime Environment (JRE) version 8 or later. ThingWorx Foundation recommends the full Development Kit (JDK™) version 11, which may require a Java SE subscription or another Oracle license for production or commercial environments. Follow the steps below to install and configure Java JDK version 11.0.12.

- Download and install the required version of Java as directed by PTC.
- Confirm the Windows environment variables are configured properly.
 - Locate your Java installation directory and copy the path. The default path is C:\Program Files\Java\jdk-<version number>.
 - Access the Windows Control Panel > System and Security > System > Advanced System Properties.
 - Click the Environment Variables button.
 - In the System variables section, verify the JAVA_HOME or JRE HOME variable was created. If absent, then create the variable as applicable to your Java platform.



- Edit the "Path" variable. Verify both the Java and the PostgreSQL paths are defined.



4. If absent, then create both the Java and the PostgreSQL variables as applicable to your platform versions.

ThingWorx Foundation

The ThingWorx Foundation Installer is the preferred method, it guides you through a new installation and automatically installs the Apache Tomcat web server. Manual mode installation is for advanced users. This example uses the Installer method.

1. Run the installer executable named "ThingWorxFoundationPostgres-9.x.x-x64.exe".

Follow the configuration steps as described in the following table.

From Location	Action
Welcome	The installer guides you through an upgrade to your existing ThingWorx Foundation installation or a new installation of ThingWorx Foundation.
Getting ready to install	<ul style="list-style-type: none"> • Prerequisites for Installation: • Your environment must meet the prerequisites for installation. For more information about these prerequisites, see the ThingWorx Foundation Installer help center topic. • You must create your ThingWorx database in the PostgreSQL database before proceeding with the installation. You need to enter the PostgreSQL database connection information, including the ThingWorx database name and credentials during installation.
Applications in this installer	<p>This tool installs the following applications on your system. They are required for ThingWorx:</p> <ul style="list-style-type: none"> • Apache Tomcat 9.0.x • ThingWorx Foundation 9.x.x <p>For more information about these prerequisites, see the ThingWorx Foundation Installer help center topic.</p>
License Agreement	<p>Read the following License Agreement. You must accept the terms of this agreement before continuing with the installation.</p> <p>Select Agreement.</p>
MLDC Agreements	<p>ThingWorx is configured to automatically transmit license usage data to PTC for license compliance purposes, thus providing you with license consumption insights and trends to best manage your license entitlements.</p> <p>Select I have read the above statement.</p>
Installation Location	<p>Please specify where to install the ThingWorx Foundation for PostgreSQL.</p> <p>The software will be installed in the following location. (default) C:\Program Files (x86)\ThingWorx</p>
Set a ThingWorx Administrator user password	Set your password for the ThingWorx Administrator Account.
Properties for ThingWorx Foundation	<ul style="list-style-type: none"> • ThingWorx HTTP Port: (default) 8080 • Java Initial Heap: (default) 1024 • Java Max Heap: (default) 2048 <p>For ThingWorx Foundation versions 8.5 and later, using SSL/TLS is the default and, unless you elect to disable it, you must properly configure it to use ThingWorx Foundation.</p> <p>Check or uncheck the option "Use SSL/TLS for a secure connection" and complete the following if checked.</p> <ul style="list-style-type: none"> • ThingWorx HTTPS Port: (default) 8443 • Tomcat Keystore File (.jks file): user supplied • Tomcat Keystore Password: user supplied • Tomcat Key Pair Alias Name: user supplied. • Tomcat Key Pair Alias Password: user supplied.
PostgreSQL Connection Information	<p>Enter your PostgreSQL connection information here:</p> <ul style="list-style-type: none"> • IP address or Host Name: (example) localhost • Port number: (default) 5432 <p>Enter your ThingWorx database information.</p> <ul style="list-style-type: none"> • ThingWorx Database Name: (example) thingworx <p>Enter database credentials.</p> <ul style="list-style-type: none"> • Username: (example) twadmin • Password: user supplied
Configuration Summary	<p>You are about to install ThingWorx Foundation for PostgreSQL.</p> <p>Review the information about your installation.</p>
Installing ThingWorx Foundation for PostgreSQL	Setup is installing ThingWorx Foundation for PostgreSQL. This might take a few minutes.
Succeeded ThingWorx Foundation for PostgreSQL installation	<p>Setup has finished installing ThingWorx Foundation for PostgreSQL.</p> <p>Launch License Configurator</p> <p>Important: It is recommended to complete licensing now.</p>

Microsoft JDBC Driver 9.4

Microsoft JDBC Drivers for SQL Server are required for the Apache Tomcat 9.0 ThingWorx-Foundation to connect to a SQL Server database. The Microsoft JDBC Driver provides different Java files to be used in correspondence with your preferred Java Runtime Environment (JRE) settings. To simplify configuration, the process library redistributes the JDBC driver files for the Tomcat 9.0 library.

1. Navigate to the process library directory Templates > PTC Content > 6. JDBC Drivers and extract the file that is named JDBC.zip.
2. Copy only the 9.4 files into the following location: C:\Program Files (x86)\ThingWorxFoundation\tomcat\apache- tomcat-9.0.xx\lib
 - mssql-jdbc-9.4.1.jre8.jar
 - mssql-jdbc-9.4.1.jre11.jar
 - mssql-jdbc-9.4.1.jre16.jar

Enable Extension Import

Extensions are a core aspect of the architecture and design of ThingWorx. Partners, third-parties, and users can add new functionality into the system in a seamless manner. Extensions can be service (function/method) libraries, connector templates, widgets, and more.

Extension import is disabled by default for all users but can be enabled by configuring the platform-settings.json file.

1. Use Windows explorer to browse the directory: C:\ProgramFiles(x86)\ThingWorxFoundation\ThingworxPlatform
2. Open the platform-settings.json file with a text editor, such as Notepad.
3. Add the following ExtensionPackageImportPolicy parameters within the scope of the PlatformSettingsConfig section.

```
{,
  "ExtensionPackageImportPolicy": {
    "importEnabled": true,
    "allowJarResources": true,
    "allowJavascriptResources": true,
    "allowCSSResources": true,
    "allowJSONResources": true,
    "allowWebAppResources": true,
    "allowEntities": true,
    "allowExtensibleEntities": true
  }
}
```

4. Save and close the platform-settings.json file.
5. Restart the ThingWorx Server.

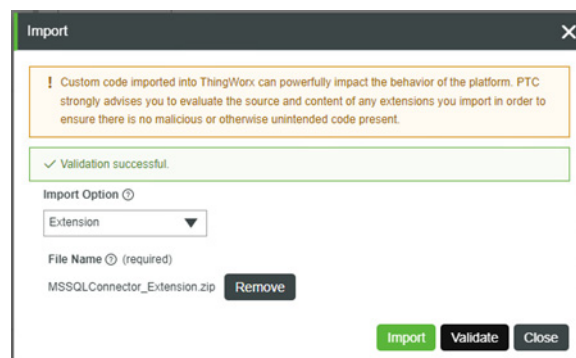
Import Extensions

The process library provides zip archives consisting of extension files and a metadata file. To install the library extensions into ThingWorx, import the zip files as described in the following steps.

1. Use Windows explorer to navigate to the Process Library > Templates > PTC Content > 5. ThingWorx Extensions > Extensions directory.
2. Extract the “Extensions.zip” file to expose the required extensions.
3. Open ThingWorx Composer™ using a web browser following the URL examples with your host name and port that is designated during installation.

<http://localhost:8080/Thingworx/Composer> or <https://localhost:8443/Thingworx/Composer>

4. Enter the ThingWorx Administrator account username/password designated during installation.
5. Select the up/down arrow button on the left menu and select Import.
6. Change the import option to “Extension”.
7. Select the “browse” button and navigate the Process Library > Templates > PTC Content > 5. ThingWorx Extensions > Extensions
8. Select the MSSQLConnector_Extension.zip file and click “Validate”.
9. If the validation fails, confirm that the platform-settings.json file was correctly formatted per the section above.
10. If the validation is successful, select “Import”.



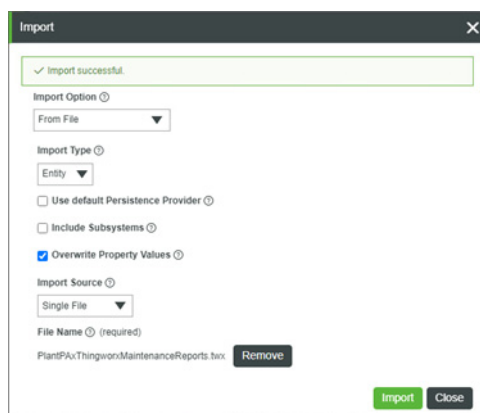
11. Repeat steps 5 ...10 to import the additional extensions provided with the Process Library.
 - custom-charts_ExtensionPackage-3.0.3.zip
 - d3Pie.zip
 - MED-61341-CD-092_F000_CSV-Parser-Extensions-4-0-2.zip
 - MED-61342-CD-092_F000_Custom-Charts-Extension-Package-3-0-5.zip

IMPORTANT For security purposes, after successfully importing the extensions, change all parameters back to “false” within the platform-settings.json file. Then restart the platform to disable any extensions from being imported.

Import PlantPax Projects

The process library contains ThingWorx Foundation templates for PlantPax Maintenance Reports and Application Launcher projects. The project templates are provided in compressed binary format (.twx) files to import into ThingWorx.

1. Using ThingWorx Composer, select the up/down arrow button on the left menu and select Import.
2. Use your host name and port that is designated during installation.
http://localhost:8080/Thingworx/Composer or https://localhost:8443/Thingworx/Composer
3. Change the import option to “File”.
4. Select the “browse” button and navigate the Process Library > Templates > PTC Content > 2. PlantPaxThingWorxMaintenanceReports
5. Select “ThingWorxMaintenanceReports.twx” and keep default import options.
6. Select Import. This operation may take several minutes to complete.



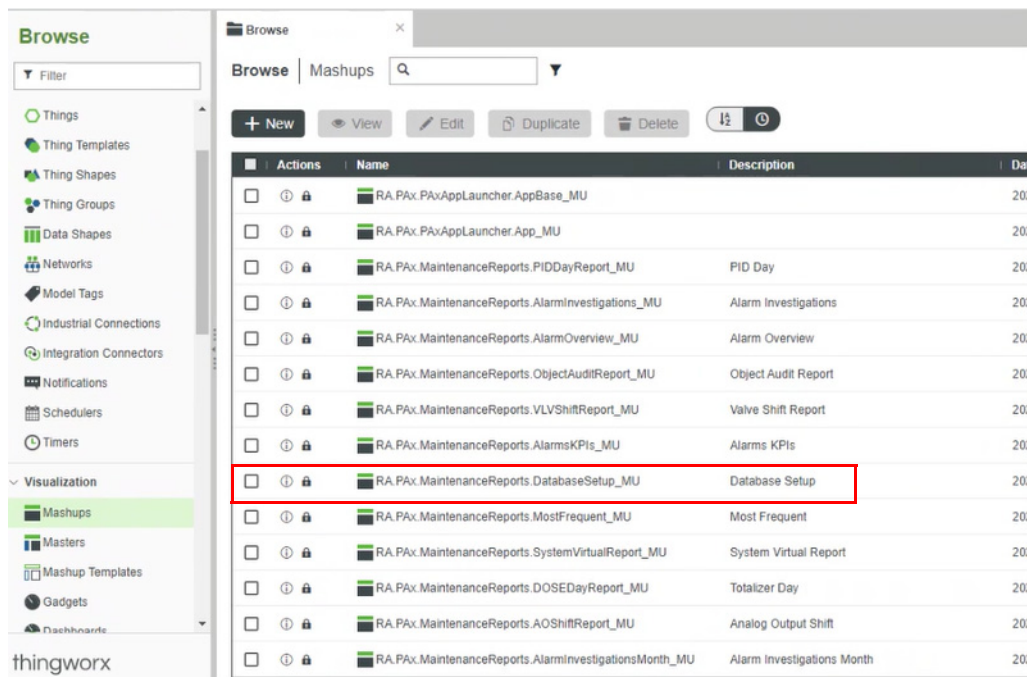
7. Select the up/down arrow button on the left menu and select Import again.
8. Change the import option to “File”.
9. Select the “browse” button and navigate the Process Library > Templates > PTC Content > 1. PlantPaxAppLauncher
10. Select “PlantPaxAppLauncher.twx” and keep default import options.
11. Select Import.

PlantPax Project Database Setup

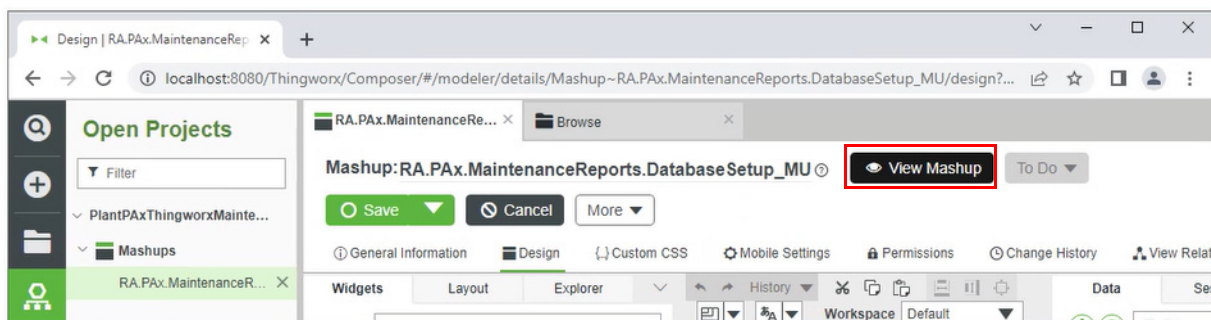
The PlantPax Maintenance Reports and Application Launcher projects require customizations to align with your PlantPax system. The following procedure uses ThingWorx Composer to change the database setup configuration, so it matches your environment.

1. ThingWorx Composer is accessed using a supported web browser. Type the URL based on this example format.
2. Use your host name and port that is designated during installation.
http://localhost:8080/Thingworx/Composer or https://localhost:8443/Thingworx/Composer

3. Select the “browse” folder on the left menu, then select “Mashups” under the Visualization section.



4. Select “RA.PAx.MaintenanceReports.DatabaseSetup_ME” and the mashup opens in a new tab.
5. Select the “View Mashup” button and the Database Setup webpage will open in a new browser tab.



6. Select the new browser tab and enter your Database Setup for both Alarms and Events Database and Process Object and System Reports.
7. Use the table below to enter your PlantPax database system parameters into the database setup fields.

Parameter	Description
JDBC Driver Class Name	Default Class Name: com.microsoft.sqlserver.jdbc.SQLServerDriver
JDBC Connection Strings	<p>This parameter directs the ThingWorx server to the SQL Server containing your PlantPax system databases.</p> <p>Default format: jdbc:sqlserver://"IP address";databaseName="yourDBname"</p> <ul style="list-style-type: none"> IP address of SQL Server = Host IP address of your SQL Server computer (exclude quotes) youDBname = name of the Alarms and Events database (exclude quotes) youDBname = name of the Centralized Database (exclude quotes) <p>Note: See the example in Step 9.</p>
connectionValidationString	Default: SELECT GETDATE()
Maximum Number of Connections	Default: 4
SQL Database User Name	User specific
SQL Database Password	User specific

8. Press “Submit Configuration” buttons at the bottom of each database after you have entered your parameters.
9. The connection status displays a green light for a successful configuration. For unsuccessful configuration, See [Troubleshooting on page 117](#).

The following example displays a configuration for a SQL server assigned IP address 192.168.1.80 that contains databases that are named FTAE_DB for Alarms and Events and AssetFrameworkDB for the Centralized database.

Database Setup for Alarms and Events Reports

JDBC Driver Class Name

JDBC Connection String

connectionValidationString

Maximum Number of Connections

Database User Name

Database Password

Confirm Password

Submit Configuration

Congratulations. Database configured successfully

Database Setup for Process Object and System Reports

JDBC Driver Class Name

JDBC Connection String

connectionValidationString

Maximum Number of Connections

Database User Name

Database Password

Confirm Password

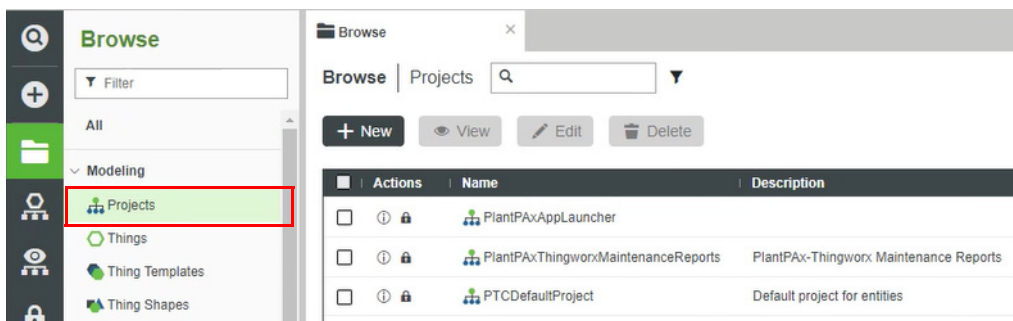
Submit Configuration

Congratulations. Database configured successfully

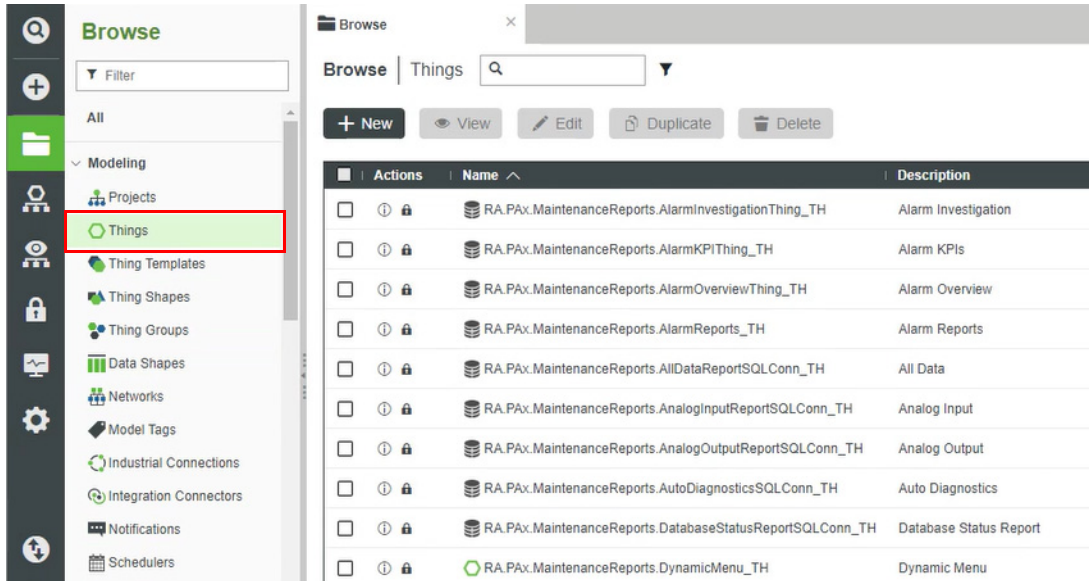
PlantPax Project Structure

ThingWorx Foundation projects are designed using the ThingWorx Composer via a web browser. The following section explains where the PlantPax project components can be accessed and modified.

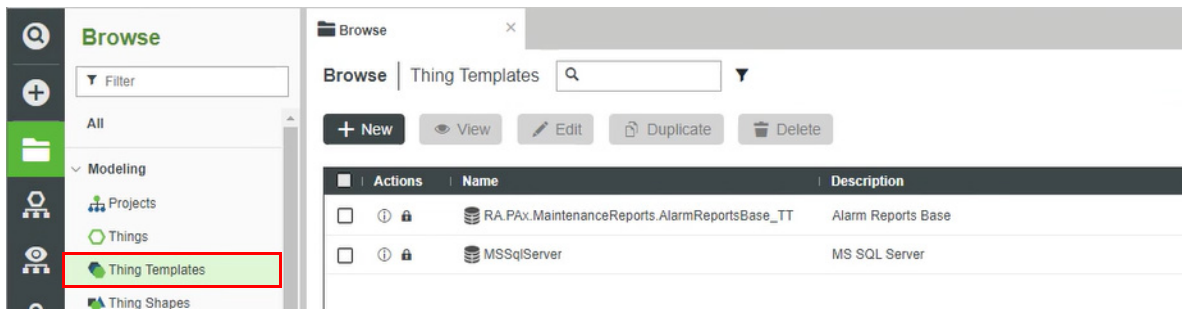
1. Select the browse folder on the left and selecting "Projects" under modeling assets. Here you can view the two imported projects PlantPax Maintenance Reports and Application Launcher.



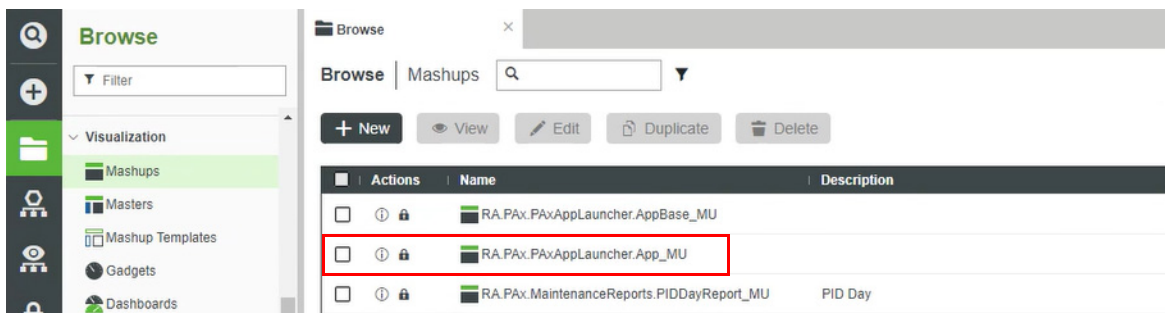
2. Select “Things” under modeling assets. Things inherit functionality from Thing Templates. Here you can view individual Things that are used within PlantPAx Maintenance Reports.



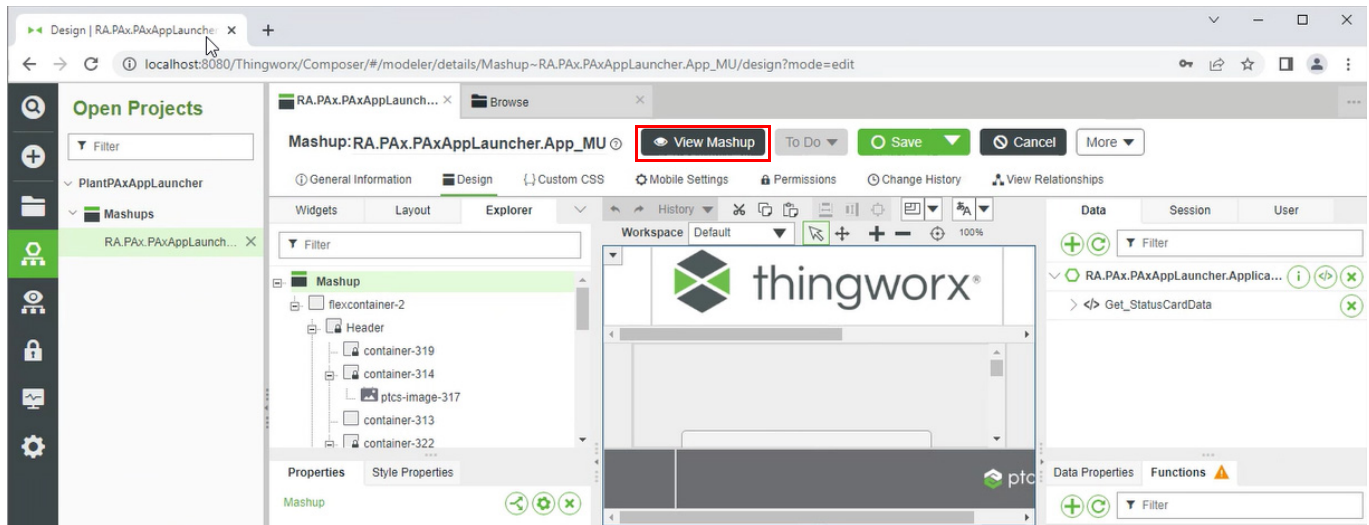
3. Select “Thing Templates” under modeling assets. Thing Templates provide the base functionality to Things with their properties, services, events, and subscriptions.



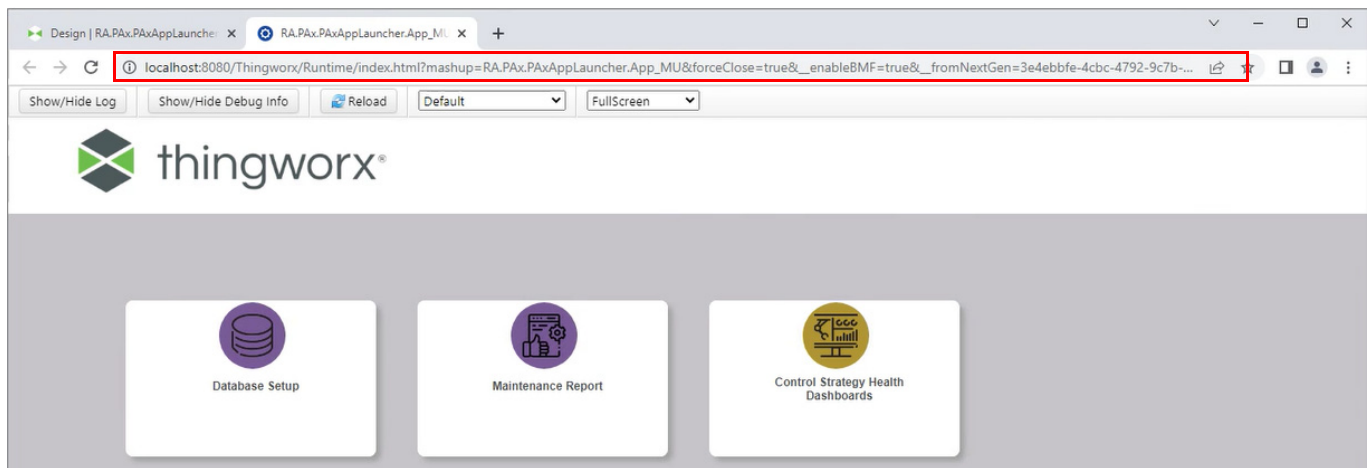
4. Select “Mashups” under visualization assets. This is where the Database Setup mashup used earlier is located. Now find the mashup named “RA.Pax.PAxAppLauncher.App_MU” and select to open it.



5. The mashup “RA.Pax.PAxAppLauncher.App_MU” will open in a new tab. On the new tab, select the “View Mashup” button.



If the Database Setup performed earlier was successful, the AppLauncher webpage displays three boxes. There is an option to modify the Database Setup, an option to view Maintenance Reports, and an option for Control Strategy Health Dashboards.



Copy the AppLauncher Mashup URL and save it. This is the unique URL that you provide to your plant operators and managers that use the ThingWorx PlantPax content.

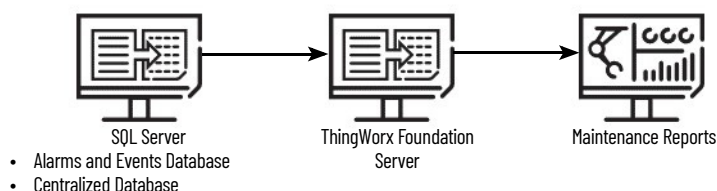
If the page only displays the ThingWorx logo, then the connection to the SQL databases likely didn't succeed. See [Troubleshooting on page 117](#).

ThingWorx Maintenance Reports

PTC® ThingWorx® Maintenance Reports are an enterprise-level reporting option for a PlantPax® 5.0 system. You can view the same type of content that is included with the Standard Reports offering, along with expanded connectivity to other enterprise-level applications and platforms.

About

The ThingWorx Maintenance reports display content in a visual format of graphs and charts. This enhanced layout provides PlantPax operators and maintenance engineers quick and easy insight into plant operations. To start viewing the ThingWorx Maintenance, open your web browser and type in the AppLauncher Mashup URL saved from the prior section. Select the Maintenance report button from the homepage.



Alarms and Events

The PlantPax Alarms and Events ThingWorx Reports are designed to provide users with insights into the health of their PlantPax alarm system. These reports are built within ThingWorx and function out of the box simply by connecting to your existing Alarms and Events Database within the system's SQL Server. These reports are expected to be added to an existing ThingWorx deployment. Functionality includes an alarm overview dashboard, an alarm KPI dashboard and an extensive set of reports. Users can access these reports within the DCS environment either from an internet browser or embedded in FactoryTalk® View SE displays. These reports are intended to help End Users comply with the alarm management lifecycle that is documented in the ANSI/ISA-18.2-2016 standard and similar guidelines and recommendations.

Report Type	Details
Alarm Overview	Total Alarm Count
	Alarm Count by State
	Alarm Count by Priority
	Alarm Count by Area
	Alarm Count by Class
Alarm KPIs	Fleeting Alarm Count
	Chattering Alarm Count
	Average Alarm Rate / 10 Min
	Average Alarm Rate / Hour
	Average Alarm Rate / Day
	% 10-minutes periods containing more than 10 alarms
	Time in Flood
	Top 10 Bad Actors%

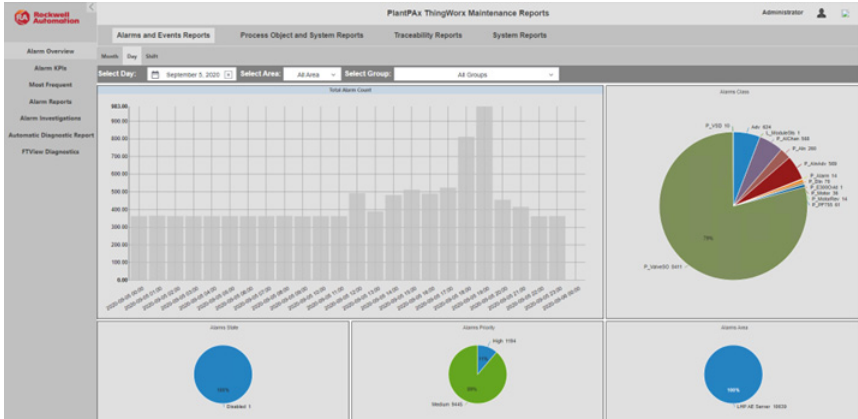
Report Type	Details
Most Frequent Alarms	Top Alarms by Duration
	Top Alarms by Disabled Duration
	Top Alarms by Shelved Duration
	Top Alarms by Suppressed Duration
	Top Alarms by Time to Acknowledge
	Top Alarms by Frequency
	Top Alarms by Shelved Frequency
	Top Alarms by Disabled Frequency
	Top Alarms by Suppressed Frequency
Alarm Reports	Alarms by Class
	Alarms by Duration
	Alarms by Priority
	Alarms by Severity
	Alarms by Disabled
	Alarms by Shelved
	Alarms by Stale
	Alarms by Suppressed
Alarm Investigations	Top Alarms by Duration
	Top Alarms by Disabled Duration
	Top Alarms by Shelved Duration
	Top Alarms by Suppressed Duration
	Top Alarms by Time to Acknowledge
	Top Alarms by Frequency
	Top Alarms by Shelved Frequency
	Top Alarms by Disabled Frequency
	Top Alarms by Suppressed Frequency

Alarm Report Navigation

The Alarm Overview is the home page for the dashboards. To populate the initial dashboard, select the entries that you want, in this order:

- 1. Select the area
- 2. Select the group
- 3. Select the period

Use previous and next buttons to see previous or next reports of the selected period. The report shows data with a maximum of one month.

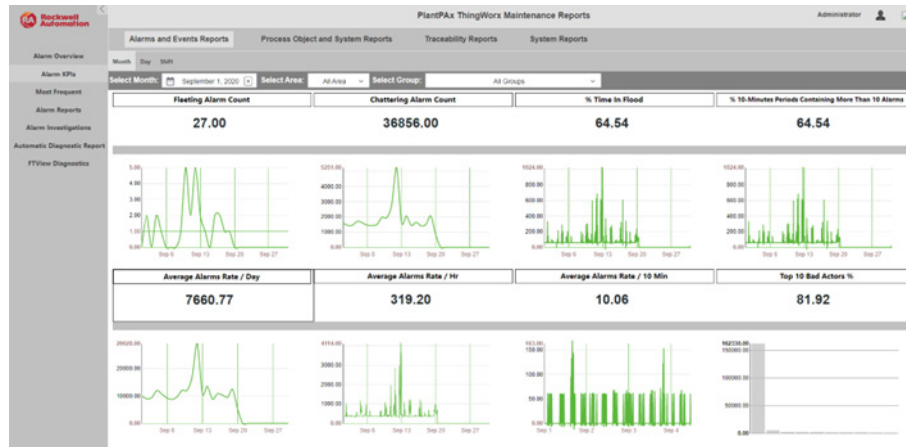


Alarm KPI

To populate the Alarms KPI dashboard, select the entries that you want, in this order:

1. Select the area
2. Select the group
3. Select the period

Use previous and next buttons to see previous or next reports of the selected period. The report shows data with a maximum time period of one month.

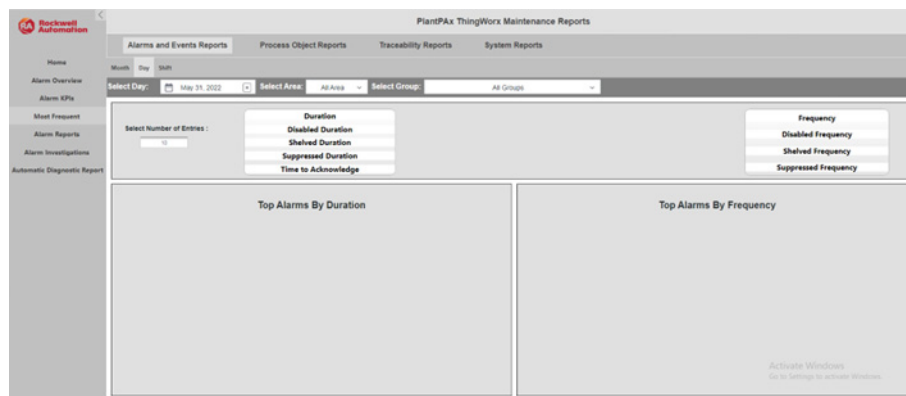


Most Frequent Alarms

To populate the Most Frequent Alarms dashboard, select the entries that you want, in this order:

1. Select the area
2. Select the group
3. Select the number of entries that you want to populate (1...25)
4. Select the period

Use previous and next buttons to see previous or next reports of the selected period. The report shows data with a maximum of one month.



Alarm Reports

To populate the Alarm Reports dashboard, select the entries that you want, in this order:

- 1. Select the area
- 2. Select the group
- 3. Select a report from the available drop-down list
- 4. Select the period

Use previous and next buttons to see previous or next reports of the selected period. The report shows data with a maximum of one month.

Rockwell Automation

Home

Alarm Overview

Alarm KPIs

Most Frequent

Alarm Reports

Alarm Investigations

Automatic Diagnostic Report

PlantPax ThingWorx Maintenance Reports

Alarms and Events Reports

Process Object Reports

Traceability Reports

System Reports

Month

Day

Week

Select Month

May 1, 2022

Select Area

FactoryTalk

Select Group

A1000_Startup

Select Report

AlarmAndAlarmClass

Name	Action Time	Reaction Time	Class	Location	Message
PLCProgram:AT1000_Startup	2022-05-01 12:37:32		AT1000_Startup	1488-10-05	AT1000_Startup: Equipment Module: Internal Trip
PLCProgram:AT1000_Startup	2022-05-19 19:30:34		AT1000_Startup	1488-10-05	AT1000_Startup: Equipment Module: Internal Trip
PLCProgram:AT1000_Startup	2022-05-19 12:38:22		AT1000_Startup	1488-10-05	AT1000_Startup: Equipment Module: Internal Trip
PLCProgram:AT1000_Startup	2022-05-19 12:38:38		AT1000_Startup	1488-10-05	AT1000_Startup: Equipment Module: Internal Trip
PLCProgram:AT1000_Startup	2022-05-19 12:42:48		AT1000_Startup	1488-10-05	AT1000_Startup: Equipment Module: Internal Trip
PLCProgram:AT1000_Startup	2022-05-21 15:53:22		AT1000_Startup	1488-10-05	AT1000_Startup: Equipment Module: Internal Trip
PLCProgram:AT1000_Startup	2022-05-21 15:53:22	2022-05-21 15:53:22	AT1000_Startup	1488-10-05	AT1000_Startup: Equipment Module: Internal Trip
PLCProgram:AT1000_Startup	2022-05-19 12:38:21	2022-05-19 12:38:21	AT1000_Startup	1488-10-05	AT1000_Startup: Equipment Module: Internal Trip

Activate Windows
Go to Settings to activate Windows.

Alarm Investigations

To populate the Alarm Reports dashboard, select the entries that you want, in this order:

- 1. Select the area
- 2. Select the group
- 3. Select a report from the available drop-down list
- 4. Designate the number of entries that you want in the table
- 5. Select the period

Rockwell Automation

PlantPax ThingWorx Maintenance Reports

Alarms and Events Reports

Process Object Reports

Traceability Reports

System Reports

Home

Alarm Overview

Alarm KPIs

Most Frequent

Alarm Reports

Alarm Investigations

Automatic Diagnostic Report

Month: May 28, 2022

Day: 10/1

Select Area: Select Area

Select Group: Select Group

Select Report: TopAlarmAndAlarmClass

Entries: 100

Alarm

Reaction Time

Reaction Time

Alarm Status

Reaction Duration

Activate Windows
Go to Settings to activate Windows.

Process Objects and System

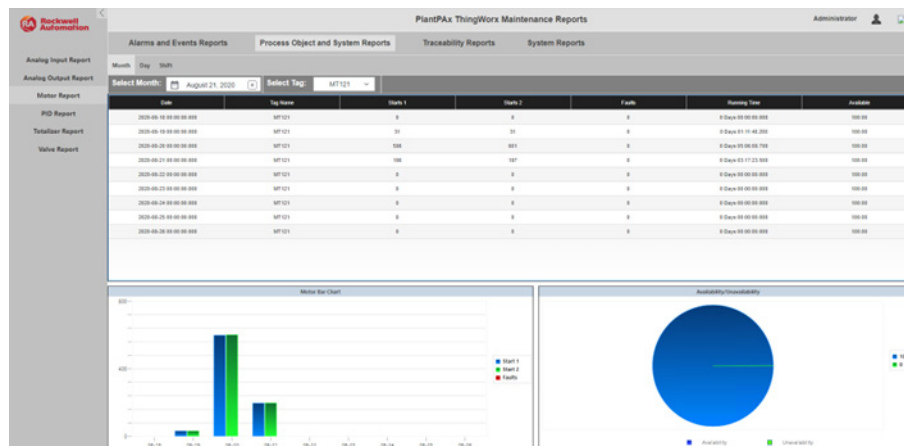
The PlantPAx ThingWorx Process Objects and System Reports are built for ThingWorx and function by connecting to the PlantPAx Centralized Database. See "PlantPAx Centralized Database" section for more information. These reports are meant to be used by maintenance personnel to maintain their process plant operations. The reports utilize FactoryTalk AssetCentre, FactoryTalk Alarms and Events and FactoryTalk Historian for data collection. Asset Framework processes the Historian data while generating Event Frames from Process Events.

Report	Description
Process Object Reports	Shows information regarding each control module that is found throughout the plant. The data for these reports is logged at the end of every hour. Each control module reporting consists of three reports, which start with a Monthly report and drill down into day and shift reports. <ul style="list-style-type: none"> • A Monthly Report which shows the data for every day in the month. • A Daily Report that shows the data for every shift in a selected day. Starting from the start of the days first shift and ending with the final shift. • Shift Report that shows the hourly data that is logged for a selected shift.
Traceability Reports	Shows information relating to accountability and events.
System Reports	Shows diagnostic information of the current state of the reporting system.

Process Object Report Navigation

To populate the Process Object Reports dashboard, select the entries that you want, in this order:

1. Object Report that you want to view
2. Select the Report Time Period, Month/Day/Shift
3. Select the Month Day or Shift
4. Select the Tag that you want to view



Traceability Report Navigation

To populate the Traceability Reports dashboard, select the entries that you want, in this order:

1. Select Traceability Report that you want to view
2. Select the Various options to filter the report based on your point of interest

[illegible]

System Report Navigation

To populate the Alarm Reports dashboard, select the entries that you want, in this order:

1. Select System Report that you want to view
2. Apply Filters to the data

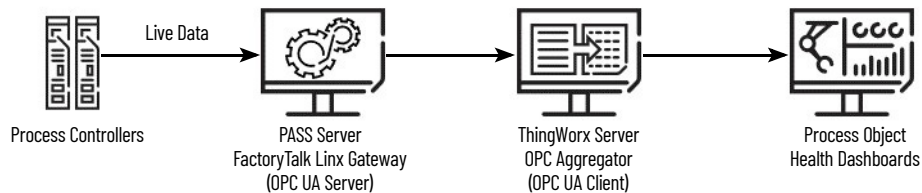
PlantPAx ThingWorx Maintenance Reports				
Alarms and Events Reports	Process Object Reports	Traceability Reports	System Reports	
	IME		Start Time	End Time
Top Logger Rawix Report	Overhaul		20:00:00	00:00:00
Top Logger Detailed Report	Afternoon		14:00:00	22:00:00
Database Status Report	Night		22:00:00	+1 Day 00:00:00
Shift Configuration Report				

Control Strategy Health Dashboards

PlantPAx® Control Strategy Health Dashboards are an enterprise-level dynamic ThingWorx® display option for a PlantPAx 5.0 system.

About

The Health Dashboards consist of a series of cards that show Control Strategy process object key values and will highlight any errors with live data in real time. The Health Dashboards automatically float the most important cards, such as ControlStrategies with activealarms, to the attention of the user for quick investigation.



Installation

OPC server and OPC client software are required to provide controller live data to the ThingWorx server. The FactoryTalk® Linx Gateway software is used for OPC UA server functionality and PTC® OPC Aggregator software is used for OPC UA client functionality. This section provides the installation and configuration steps to set up the PlantPAx Control Strategy Health Dashboards with a ThingWorx server.

FactoryTalk Linx Gateway

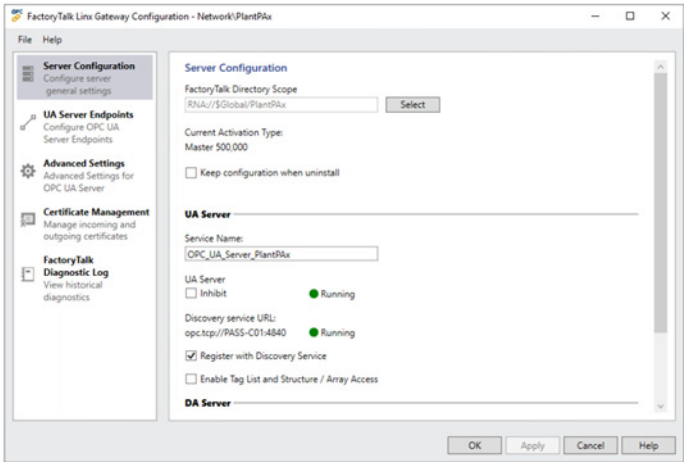
FactoryTalk Linx Gateway software is typically installed on the Process Automation System Server (PASS) that is running FactoryTalk Linx software as a data server. Use the following steps to configure FactoryTalk Linx Gateway as an OPC UA server.

1. On the computer designated for FactoryTalk Linx Gateway, verify that the software is installed and licensed.
2. Use the Windows® start menu to launch the FactoryTalk Linx Gateway Configuration.
3. Create an OPC UA server using the parameters from the following table.

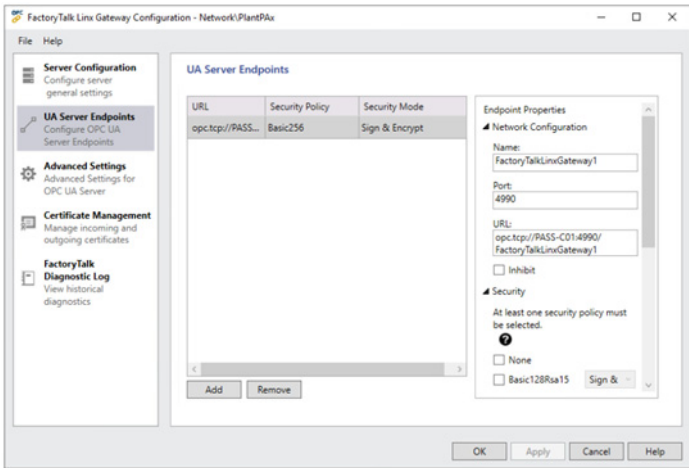
From Location	Action
Server Configuration	Select the FactoryTalk Directory Scope. The select button allows you to choose a network or local FactoryTalk directory and the project scope you want to source data from. (Example: RNA://\$Global/PlantPAx)
	Verify the current activation type to achieve gateway functionality.
	Verify that UA Server contains a Service Name and is registered with Discovery Service. (Example: OPC-UA_Server_Plant_A)
	The DA Server is optional and can be inhibited if not required for other clients.
UA Server Endpoints	Add the endpoint properties. <ul style="list-style-type: none"> • Name: (Example: FactoryTalkLinxGateway1) • Port: (Example: 4990) • URL: (Example: opc.tcp://PASS-C01:4990/FactoryTalkLinxGateway1) • Security: (Example: None) • Authentication Settings: (Example: Anonymous, Enable read access to CIA message, Enable write access to CIA message) • Diagnostic Logging: optional

From Location	Action
Certificate Management	<p>Use Certificate Management to view FactoryTalk Linx Gateway OPC UA server certificate information.</p> <p>FactoryTalk Linx Gateway creates a default, self-signed security certificate during installation. The security certificate is stored at C:\ProgramData\Rockwell\FTLinxGateway\PKI\own\cert_ftgw_opcua_server.der on the host computer.</p> <p>A custom security certificate can also be created.</p> <p>Configure the incoming and outgoing certificate as desired.</p>

Example of a FactoryTalk Linx Gateway Server Configuration



Example of a FactoryTalk Linx Gateway UA Server Endpoints Configuration



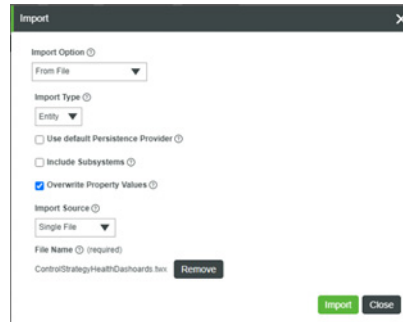
- 4. Add Windows Defender Firewall incoming rules to allow the TCP port designated for the UA Server Endpoints on the FactoryTalk Linx Gateway computer and the ThingWorx server.

Import Dashboard Templates

The Process Library contains ThingWorx Foundation templates for Control Strategy Health Dashboards. The Dashboard Templates are system-defined remote templates of type "RemoteThing". The RemoteThing type is used when creating "Things" that support OPC data sources. The project templates are provided in compressed binary format (.twx) files to import into ThingWorx. Use the following steps to import the dashboards into the ThingWorx server.

- 1. Using the ThingWorx Composer™ webpage, select the up/down arrow button on the left menu and select Import.
- 2. Change the import option to "File".
- 3. Select the "browse" button and navigate the Process Library > Templates > PTC Content > 3. ControlStrategyHealthDashboards
- 4. Select "ControlStrategyHealthDashboards.twx" and keep default import options.

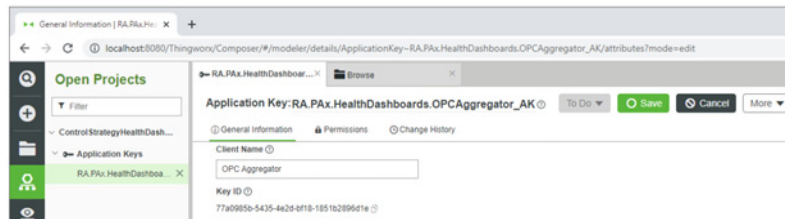
5. Select Import. This operation may take several minutes to complete.



Application Key

Application keys are security tokens encrypted in the ThingWorx database that can be used to log in to the ThingWorx application instead of using standard credentials. The key included with the health dashboards template is used to authenticate the connection to the OPC UA client. Follow the steps below to obtain the Key ID.

1. Using ThingWorx Composer webpage, select the browse button on the left menu.
2. Select Application Keys under Security Category and select RA.PAx.HealthDashboards.OPCAggregator_AK application key.
3. Scroll down the RA.PAx.HealthDashboards.OPCAggregator_AK tab and locate the Key ID.



4. Record the Key ID for use in OPC Aggregator.

OPC Aggregator

PTC ThingWorx Foundation provides OPC Aggregator software to provide OPC UA client functionality. The OPC Aggregator polls select controller tags (control strategies) and connects the live data to the ThingWorx server for Control Strategy Health Dashboards.

Download OPC Aggregator software from the PTC software portal and install on the ThingWorx server.

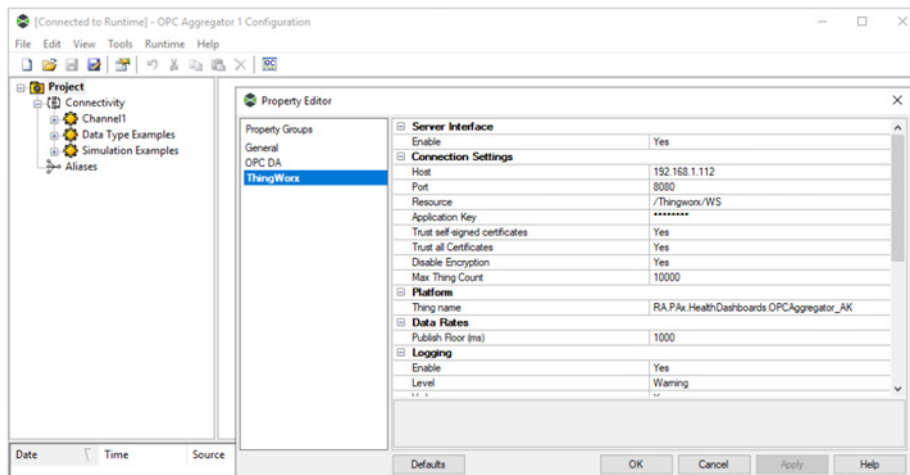
ThingWorx Connection

The first step using the OPC Aggregator is to create a connection to your ThingWorx server. Follow the steps below to create the connection using your Key ID from the previous section.

1. Use the Windows start menu to launch OPC Aggregator 1 Configuration, then double-click on "Project" in the left window.
2. In the property editor, select "ThingWorx" and change the Server Interface enable from "No" to "Yes", then modify the parameters as recommended in the following table.

ThingWorx	Action
Server Interface	Yes
Connection Settings	<ul style="list-style-type: none"> • Host: ThingWorx server IP address • Port: 8080 (or desired setting) • Application Key: copied Key ID from prior section. • Trust self-signed certificates: Yes (or desired setting) • Trust all Certificates: Yes (or desired setting) • Disable Encryption: Yes (or desired setting) • Max Thing Count: 10000
Platform	Thing name: RA.PAx.HealthDashboards.OPCAggregator_AK
Logging	<ul style="list-style-type: none"> • Enable: Yes • Level: Warning • Verbose: Yes

Example of a ThingWorx Server Connection Configuration



If the event log displays connection errors, See [Troubleshooting on page 117](#).

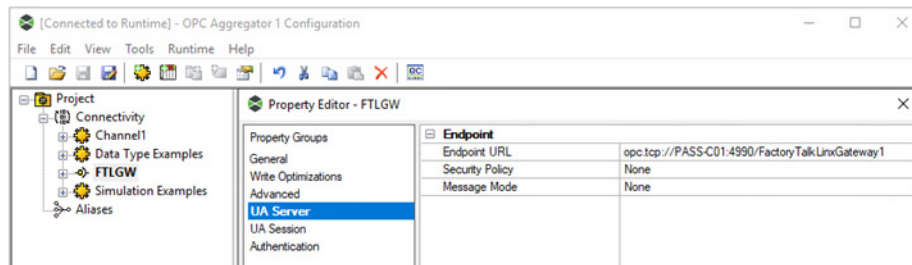
FactoryTalk Linx Gateway Connection

Once the connection to the ThingWorx server is established, the next connection to establish is to the OPC UA server (FactoryTalk Linx Gateway). Follow the steps below to set up an OPC UA client (OPC Aggregator) to OPC UA server connection.

1. From the OPC Aggregator, Select "Connectivity" in the left window. Right-click and select "New Channel" to launch the add channel wizard.
2. Create an OPC UA Client channel using the parameters from the following table.

Add Channel Wizard	Action
Select the type of channel to be created	OPC UA Client
Specify the identity of this object	Enter a name to identify the connection to your FactoryTalk Linx Gateway OPC UA server. Name: user defined (Examples: FTLGW, FactoryTalkLinxGateway1, PlantA)
Choose how write data is passed to the underlying communications driver when more than one write exists in the write queue	Optimization Method: Write Only Latest Value for All Tags (default) Duty Cycle: 10 (default)
Specify the unique URL destination of the OPC UA endpoint.	Endpoint URL: opc.tcp://hostname:portnumber/endpointname Security policy: user defined (Example: none) Message mode: user defined (Example: none) Enter the URL created in the FactoryTalk Linx Gateway configuration. (Example: opc.tcp://PASS-C01:4990/FactoryTalkLinxGateway1)
Maximum amount of time the channel should wait to successfully connect after making a connect call, and the maximum amount of time a session remains open without activity	Connection Timeout (s): 30 (default) Idle Session Timeout (min): 20 (default)
Authentication	Enter a valid account name and password to use when connecting to OPC UA endpoints that require authentication. Username: user defined (Example: none) Password: user defined (Example: none)

Example of an OPC UA Server Connection Configuration



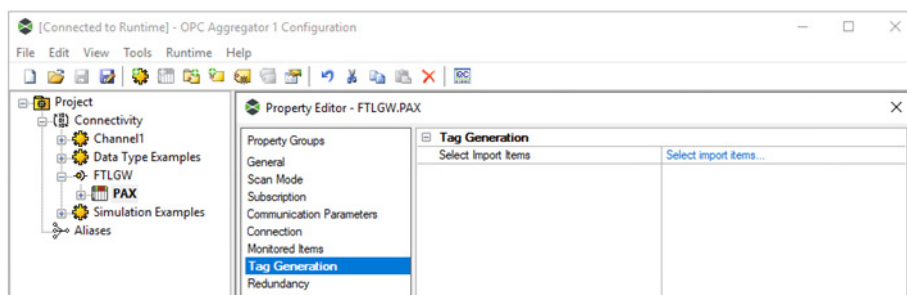
Controller Connection

Once the OPC UA client to OPC UA server connection is established, you can set up device connections to your process controllers. Use the following steps to set up a device connection to your process controller. This example configures one controller connection; however the steps can be repeated to add multiple controllers within a PlantPax system.

1. The new OPC UA Client channel provides a link to “Click to add a device.” in the left window. Select this link to launch the add device wizard.
2. Create a device connection to your process controller using the parameters from the following table.

Add Device Wizard	Action
General - Specify the identity of this object	Enter a name of one of your Process Controllers or its FactoryTalk shortcut name. Name: user defined (Example: PAX)
Scan Mode - Specify the method for determining how often tags in the device are scanned.	Scan Mode: Respect Client-Specified Scan Rate (default) Initial Updates from Cache: Disable (default)
Subscription - Specify the rate at which tags are updated by the driver Specify the maximum number of notifications the OPC UA server sends to the driver in a single published response Select the subscription method and whether device tags are registered	Publishing Interval (ms): 1000 (default) Max. Notifications per Publish: 0 (default) Update Mode: Exception (default) Registered Read/Write: (only available with Update Mode: Poll)
Connection - Specify how many times that the publishing interval can expire without the OPC UA Client Driver sending data updates or keep-alive messages before the server deletes the subscription	Lifetime Count: 60 (default) Keep-Alive Count: 5 (default)
Communication Parameters - Specify the maximum number of items in each read call to the server for subscriptions in Polled Update Mode Specify the maximum number of items in each write call to the server Specify the time allowed for each read call Specify the time allowed for each write call Choose whether an explicit read occurs after a write Specify the time that the driver waits for the initial update	Max. Items per Read: 512 (default) Max Items per Write: 512 (default) Read Timeout: 1000 (default) Write Timeout: 1000 (default) Read after Write: Enabled (default) Initial Update Timeout (ms): 5000 (default)
Monitored Items - Specify the maximum rate at which monitored items are updated Specify the number of data updates that the OPC UA server queues for the subscription Indicate if the older notification in the queue should be discarded and not sent to the driver	Sample Interval (ms): 500 (default) Queue Size: 1 (default) Discard Oldest: Enable (default)
Select the type of deadband filter to be applied to data changes	Deadband Type: None (default)
Tag Generation - Select tags from the remote OPC UA Server to auto-generate tags for this device	Select Import Items: Test the ability to browse for controller tags. After confirming the browse functionality, close the browser and use the next section to set up your tags.
Redundancy	Optional

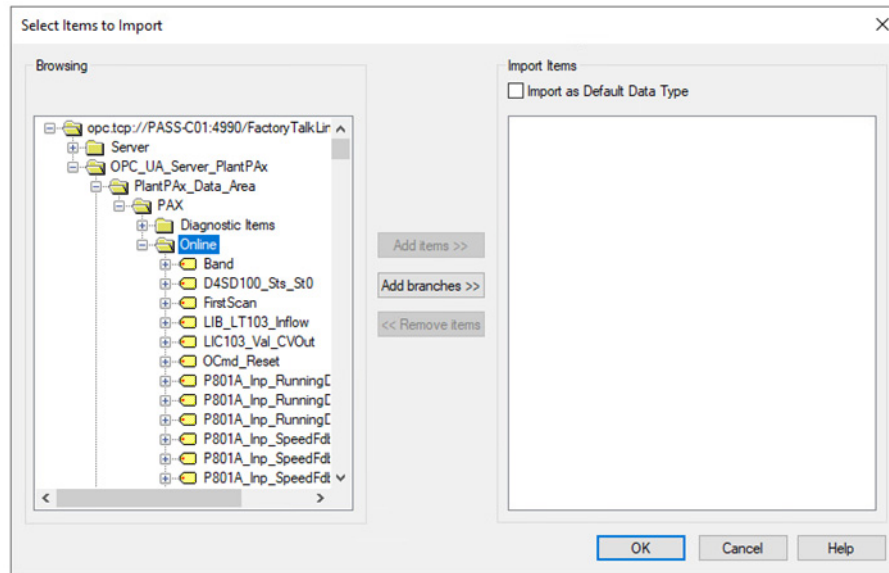
Example of a Device Connection Configuration



Tag Generation

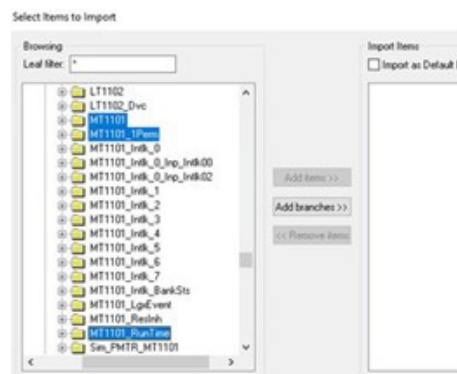
After the process controller connection is established, use the following steps for tag generation. Tag generation adds items, which are specific tags (control strategies) that the OPC UA client will put on scan. These tags will be used by the Control Strategy Health Dashboards.

1. Right-click your device and select properties to launch the Property Editor. Select the Tag Generation category and select the link to “Select import items”, which launches the tag browser.
2. Using the tag browser, expand your OPC UA server > OPC UA service name > Area > Shortcut > Online folder. The ability to browse online is an indication that the OPC UA server/client communications are good.



3. Use the browser to select objects for control strategies that you want to monitor on Control Strategy Health Dashboards. The process object and companion object are required to give full context. Hold CTRL to select multiple objects.

The following example contains a process motor (PMTR) object that is named MT1101, the motor permissive object named MT1101_1Perm and the motor statistics object named MT1101_Runtime. These objects comprise the single-speed process motor control strategy routine that is included in the process library.



4. Select your objects and select “Add Branches” to add these items to the OPC UA client.
5. Select Apply, OK, then close out the Items to import menu. The object tags are now available in the OPC Aggregator to be tied to the ThingWorx Control Strategy Things.

Thing Generation File

PlantPax Control Strategy Health Dashboards “things” are created on the ThingWorx server from CSV file imports.

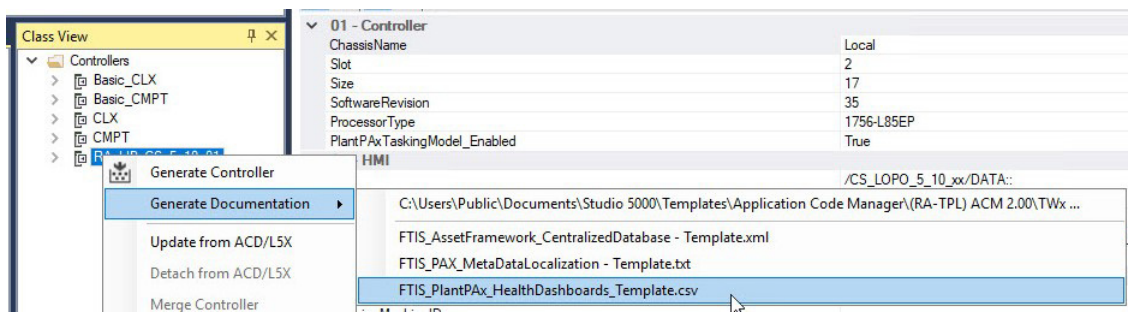
There are two methods available to create the CSV file for Dashboard things.

- ACM to generate the template file “PlantPax_HealthDashboards.csv”
- Modify the Process Library CSV template file “ControlStrategyHealthDashboards.csv”

Application Code Manager (Option 1)

ACM software is the preferred method for developing PlantPax applications and generating application content. If the controller application was created with ACM, use this method.

1. Open the ACM project.
2. In the Classic View window, right-click on the controller and select “Generate Documentation”.
3. Select the option for “PlantPax_HealthDashboards_Template.csv”



Process Library Template (Option 2)

This method is intended for controller applications that have been developed without ACM. A sample CSV has been started and distributed with the Process Library. This sample CSV contains the format that is required for ThingWorx to import and demonstrates basic examples of dashboard objects.

1. Navigate the Process Library > Templates > PTC Content > 3. ControlStrategyHealthDashboards
2. Copy the “ControlStrategyHealthDashboards.csv” file to another location on your computer and open with a CSV editor.

The screenshot shows a CSV editor window displaying the 'ControlStrategyHealthDashboards.csv' file. The table has the following columns: A (Line), B (Thing Template), C (Channel), D (Device), E (Gateway Service), F (more than Name), G (more than Name), H (Path to Data Server), I (more than Name), J (Shortcut), K (Program Scope), L (Name), M (Instrument Name EU), and N (%). The table contains 12 rows of data, each representing a different dashboard object configuration.

Line	Thing Template	Channel	Device	Gateway Service	more than Name	more than Name	Path to Data Server	more than Name	Shortcut	Program Scope	Name	Instrument Name EU
1	RA.Pax.HealthDashboards.OPCAggregator_AK_RA.Pax.DCS.IveData_CS_PAI_TT	FTISIW	FALSE	PAX	FALSE	OPC_UA_Server_Plant_A	TRUE	Area_A_Unit_A	PID	TRUE	ProgramTK301_VT101	%
2	RA.Pax.HealthDashboards.OPCAggregator_AK_RA.Pax.DCS.IveData_CS_PAI_TT	FTISIW	FALSE	PAX	FALSE	OPC_UA_Server_Plant_A	TRUE	Area_A_Unit_A	AI	FALSE	AIT300A	%
3	RA.Pax.HealthDashboards.OPCAggregator_AK_RA.Pax.DCS.IveData_CS_FAO_TT	FTISIW	FALSE	PAX	FALSE	OPC_UA_Server_Plant_A	TRUE	Area_A	AO	FALSE	XC350	%
4	RA.Pax.HealthDashboards.OPCAggregator_AK_RA.Pax.DCS.IveData_CS_PVSD_TT	FTISIW	FALSE	PAX	FALSE	OPC_UA_Server_Plant_A	TRUE	Area_B	VSD	FALSE	MT120	%
5	RA.Pax.HealthDashboards.OPCAggregator_AK_RA.Pax.DCS.IveData_CS_PVSD_TT	FTISIW	FALSE	PAX	FALSE	OPC_UA_Server_Plant_A	TRUE	Area_B	VSD	FALSE	MT130	%
6	RA.Pax.HealthDashboards.OPCAggregator_AK_RA.Pax.DCS.IveData_CS_PVSD_TT	FTISIW	FALSE	PAX	FALSE	OPC_UA_Server_Plant_A	TRUE	Area_C	Dose	FALSE	PM100	%
7	RA.Pax.HealthDashboards.OPCAggregator_AK_RA.Pax.DCS.IveData_CS_PVSD_TT	FTISIW	FALSE	PAX	FALSE	OPC_UA_Server_Plant_A	TRUE	Area_A_Unit_A	PID	FALSE	XC320	%
8	RA.Pax.HealthDashboards.OPCAggregator_AK_RA.Pax.DCS.IveData_CS_PVSD_TT	FTISIW	FALSE	PAX	FALSE	OPC_UA_Server_Plant_A	TRUE	Area_A_Unit_A	PID	FALSE	XC320	%
9	RA.Pax.HealthDashboards.OPCAggregator_AK_RA.Pax.DCS.IveData_CS_PVSD_TT	FTISIW	FALSE	PAX	FALSE	OPC_UA_Server_Plant_A	TRUE	Area_A_Unit_A	MTRE	FALSE	MT111	%
10	RA.Pax.HealthDashboards.OPCAggregator_AK_RA.Pax.DCS.IveData_CS_PVSD_TT	FTISIW	FALSE	PAX	FALSE	OPC_UA_Server_Plant_A	TRUE	Area_A_Unit_C	VIV	FALSE	XV110	%
11	RA.Pax.HealthDashboards.OPCAggregator_AK_RA.Pax.DCS.IveData_CS_PVSD_TT	FTISIW	FALSE	PAX	FALSE	OPC_UA_Server_Plant_A	TRUE	Area_A_Unit_C	VIV	FALSE	XV110	%
12	RA.Pax.HealthDashboards.OPCAggregator_AK_RA.Pax.DCS.IveData_CS_PVSD_TT	FTISIW	FALSE	PAX	FALSE	OPC_UA_Server_Plant_A	TRUE	Area_A_Unit_C	VIV	FALSE	XV110	%

3. Modify this CSV file as it pertains to your PlantPax system devices.

The following table describes the CSV file format for each column and its usage.

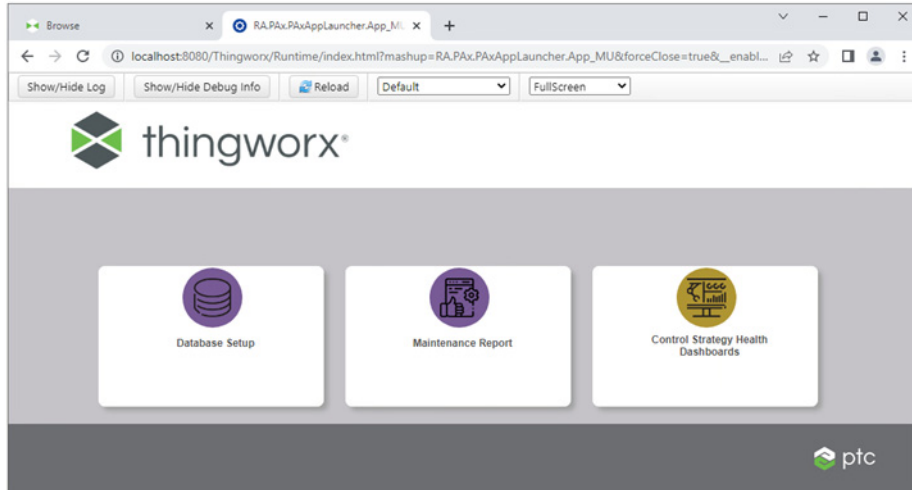
Column	Description
Line Number	Line number of each dashboard thing
Industrial Connection	Health dashboards application key that is used for ThingWorx to OPA Aggregator connection (Example: RA.Pax.HealthDashboards.OPCAggregator_AK)
Thing Template	Thing Templates for each process object type imported from the ControlStrategyHealthDashboards template <ul style="list-style-type: none"> • RA.Pax.DCSLiveData.CS_PAI_TT = Thing Template for Process Analog Input • RA.Pax.DCSLiveData.CS_PAO_TT = Thing Template for Process Analog Output • RA.Pax.DCSLiveData.CS_PDOSE_TT = Thing Template for Process Dose • RA.Pax.DCSLiveData.CS_PMTR_TT = Thing Template for Process Motor • RA.Pax.DCSLiveData.CS_PPID_TT = Thing Template for Process PID • RA.Pax.DCSLiveData.CS_PVLV_TT = Thing Template for Process Valve
Channel Name	Name given to the OPC UA channel created in OPC Aggregator configuration
Device more than one?	Device under the channel. True or False in OPC Aggregator configuration
Name	Name of the device created in OPC Aggregator configuration
Gateway Service	Gateway service that is configured in OPC Aggregator configuration More than one? True or False
Name	UA Server service name assigned in FactoryTalk Linx Gateway configuration
Path to Data Server more than one?	FactoryTalk Linx Data Server Area. True or False <ul style="list-style-type: none"> • True = If your FactoryTalk system contains one or more FactoryTalk Linx Data Server within one or more separate Areas. • False = If your FactoryTalk system contains one FactoryTalk Linx Data Server in the project root and not within an Area folder.
Name	Name of the Area containing the FactoryTalk Linx Data Server. (Example: PlantPax_Data_Area)
Shortcut Name	Device shortcut name configured in the FactoryTalk Linx Communication Setup
Program Scope	Whether the control strategy process objects for the dashboard exist in the controller scope or program scope of the Logix controller.
Name	If program scope, specify the name of the program containing the process object. (Example: Program.eTK101_Example)
Instrument Name	Controller tag name of the process object to be used on the dashboard
EU	Engineering unit symbol displayed on the dashboard

Create Dashboard Things

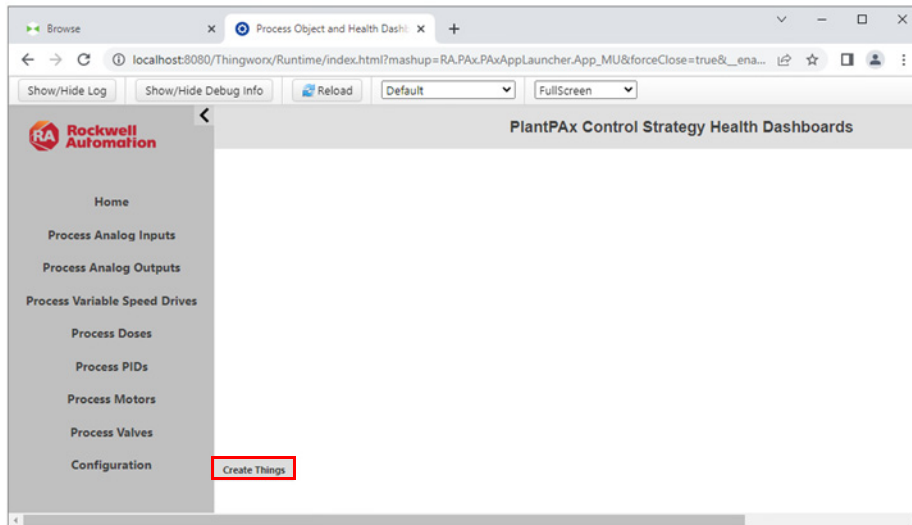
When the CSV file containing the things you want to use on health dashboards is completed, you can import the CSV file into ThingWorx.

IMPORTANT Before importing new dashboards things into ThingWorx, ensure that the OPC Aggregator has all Tags Generated for the objects in the CSV file. If not, the dashboards will not link up with the correct tags in the controller. See [Troubleshooting on page 117](#) if dashboards are not updating with live data.

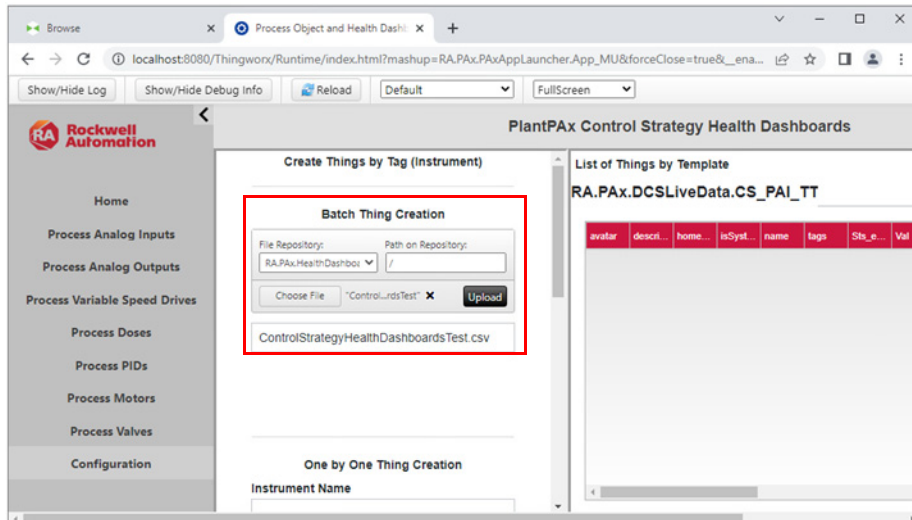
1. Open a web browser and access the App Launcher page with the three boxes (Database Setup, Maintenance Reports, and Control Strategy Health Dashboards).



2. Select the box for Control Strategy Health Dashboards
3. On the page for PlantPAx Control Strategy Health Dashboards, select Configuration > Create Things on the bottom-left menu.



4. Access the middle window, labeled "Create Things by Tag (Instrument)". To choose your CSV file, use the "Batch Thing Creation" form and select "Upload".



- After the upload success message appears, select the button to “Create Things”.

Batch Thing Creation

File Repository: RA.PAx.HealthDashbo Path on Repository: /

Choose File
“Control...rdsTest” ✕
Upload

ControlStrategyHealthDashboardsTest.csv

Create Things

- The “List of Things by Template” will be displayed in the right window. Verify that each of the templates have things created from your CSV file. Use the ellipse button at the upper right to view more thing templates.

List of Things by Template

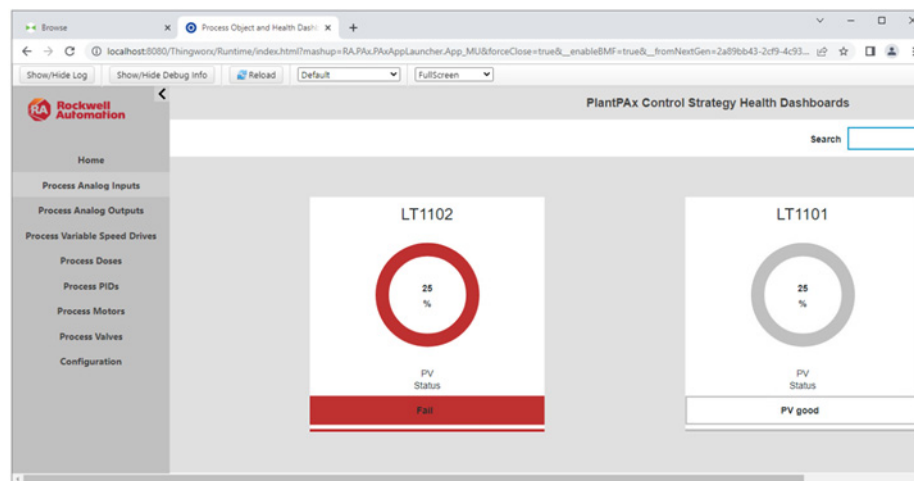
RA.PAx.DCSLiveData.CS_PA1_TT										RA.PAx.DCSLiveData.CS_PAO_TT										RA.PAx.DCSLiveData.CS_PVSD_TT									
avatar	descr.	home	isCtyd	name	tags	Sts_e	Val	isCon	Sts_Lo	isRep	ThingTemplate	Sts_Ann	Sts_H	report	Sts_Pk	Val_In	Sts_J	Val_H											
/Things/ LT1101				LT1101_RemoteThing		0	25	true		true	RA.PAx.DCSLiveData.CS_PA1_TT			0	Fri Feb 3	25		0											
/Things/ LT1102				LT1102_RemoteThing		0	25	true		true	RA.PAx.DCSLiveData.CS_PA1_TT			0	Fri Feb 3	25	true	0											

Health Dashboard Navigation

Select the various Process Object categories on the left menu to test your dashboards.

The statistics shown on several dashboards (such as running hours) are calculated in the reporting layer, thus the statics instructions are not required within the controller application for this functionality.

The following example displays Control Strategy Health Dashboards for a Level Sensors LT1101 and LT1102. The Level Sensors are controlled by Process Analog Input (PAI) instructions in the process controller. LT1101_Remote_Thing and LT1102_Remote_Thing were created using the CSV import. A simulated channel fault was generated on LT1102. Notice that LT1102 has floated to the first card to bring attention to a fault. See [Troubleshooting on page 117](#) if dashboards are not updating with expected data.

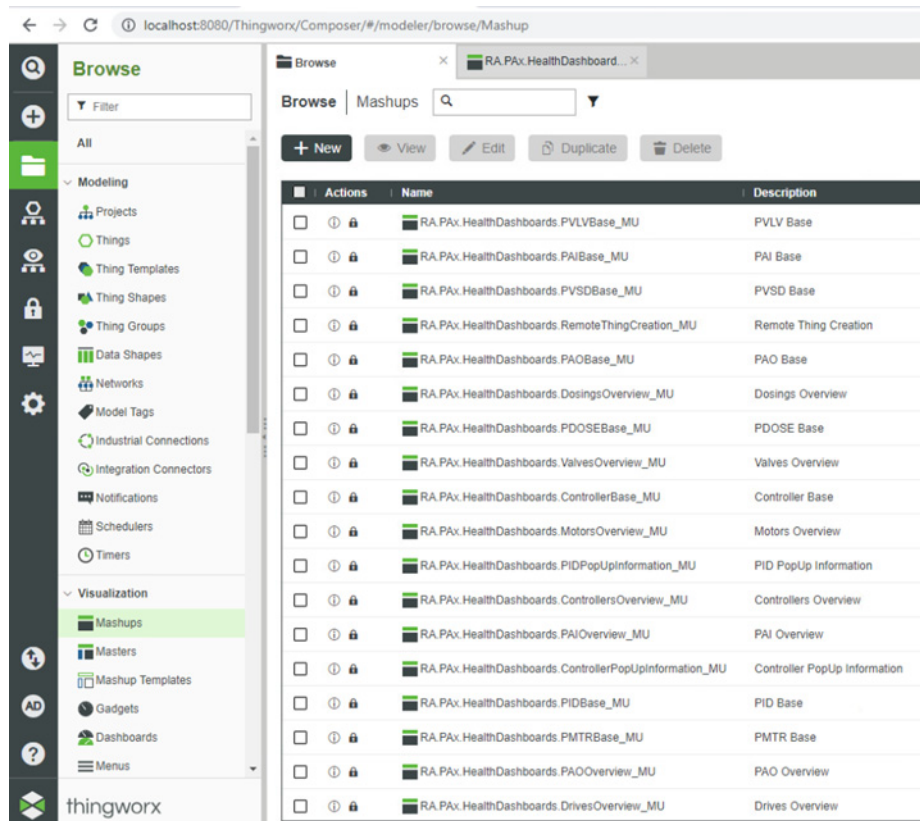


Health Dashboard Customization

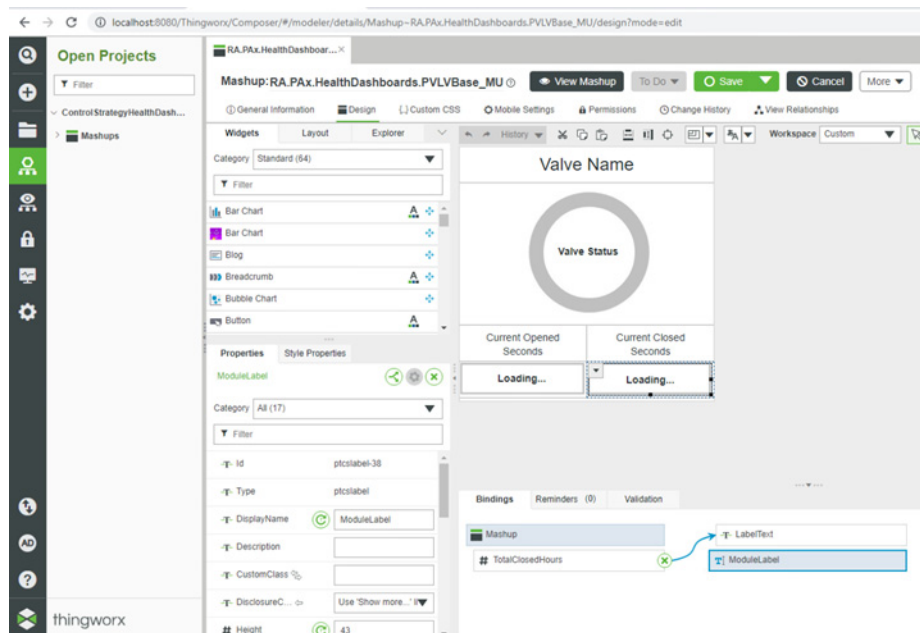
The PlantPax Control Strategy Health Dashboard Templates provide users a starting point with process object cards. The base cards display variables believed to be import for most process users. To display different variables, users can customize the card mashups using the ThingWorx Composer via a web browser. The following section shows where the PlantPax project components can be accessed and modified.

- Select the browse folder on the left and select “Mashups”.

2. Select the RA.Pax.HealthDashboards.ObjectBase_MU for the object of choice.



3. Variable Bindings and Properties can be modified accordingly to the user's requirements.

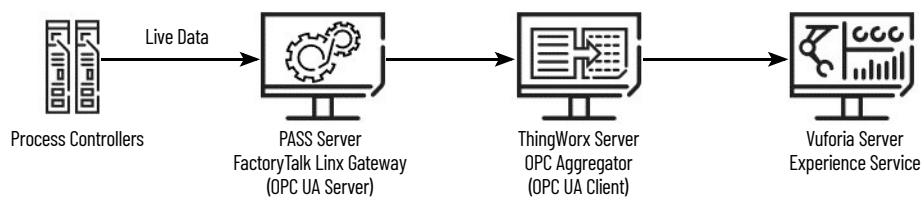


4. Save the Mashup and refresh your browser window with the open dashboard.

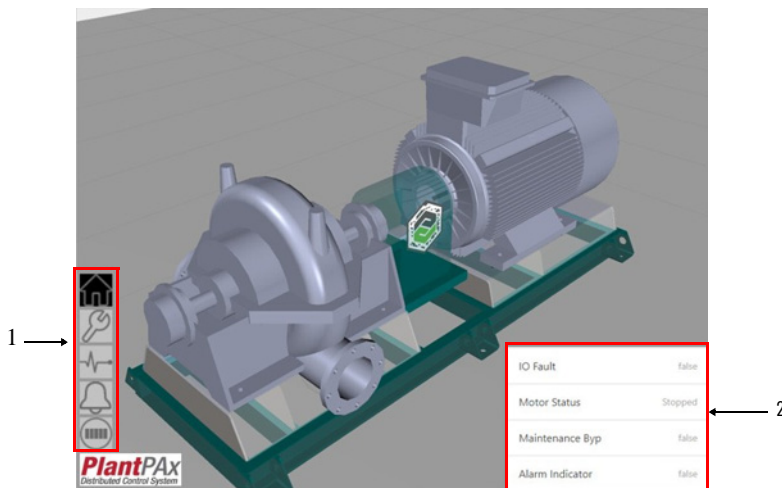
Vuforia Experiences

PTC® Vuforia® Studio is a comprehensive, scalable enterprise augmented reality (AR) platform. AR experiences are published to an Experience Service, which allows you to access them on mobile phones, tablets, or eyewear devices through the Vuforia View application. Vuforia experiences enable operators to see certain insights while on the plant floor. Vuforia AR can be useful, when maintenance personnel are not within proximity to an HMI terminal but require real-time asset information, along with installation, operation, and maintenance instructions.

About



PlantPax® offers Vuforia Control Strategy Experiences built upon the ThingWorx® Thing Templates. These experiences provide PlantPax 2D Overlay buttons that provide HMI faceplate-like navigation with corresponding status values on the other side. The generic graphical models that are provided in the templates can be replaced with actual 3D models representing your field devices. The following capture is an example of the process motor PMTR template as it would look on a tablet.



Item	Description
1	Navigation Panel
2	Popup Panel

Installation

Vuforia Experiences requires the following software components.

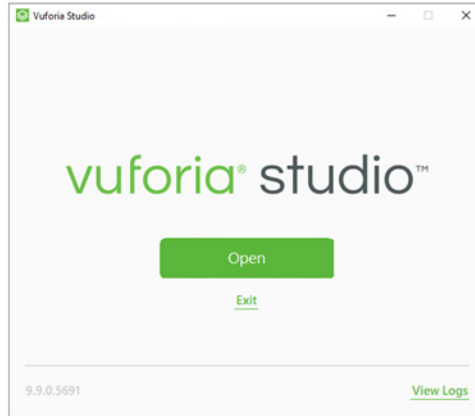
- Vuforia Studio
- Vuforia Experience Service
- Vuforia View App from Apple, Google, or Microsoft®

This section covers the installation and configuration to set up the PlantPax Vuforia Experiences with a ThingWorx server.

Vuforia Studio

Vuforia Studio - A web-native, easy-to-use tool for authoring domain and task-specific experiences that provide an integrated view of digital and physical product data, dashboards, and alerts with 2D, 3D, and augmented reality.

Vuforia Studio installation uses an online installer, which is available from the PTC software portal. Following a successful installation, a one-time application reference prompts you to “Open” Vuforia Studio.



The Vuforia Studio application reference shortcut can be found in the C:\Users\User\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\PTC Inc\ directory. Copy this shortcut to the Windows® startup folder to launch the application after Windows starts.

Vuforia Experience Service

Vuforia Experience Service is an enterprise-class, secure, and scalable server that publishes, updates, and deletes experience content that is generated through Vuforia Studio. It also identifies and delivers Experience content that is requested through Vuforia View.

ThingWorx Foundation must be installed and established before installing the Vuforia Experience Service. Use the following table as a guide for the Experience Service installation wizard.

From Location	Action
Welcome	Setup - Experience Service Welcome to the Experience Service Setup Wizard.
License Agreement.	Read and select I accept the Agreement.
Installation Directory	Specify the directory where the Experience Service is installed. Installation Directory: C:\ptc\studio-es (default)
Experience Service Configuration	Port: 2019 (default) Database Type: PostgreSQL
Database Configuration	Configure PostgreSQL connection Database Hostname: localhost Database Port: 5432 Database Name: thingworx Database Username: twadmin Database Password: ***** <input type="checkbox"/> PostgreSQL Server requires TLS
TLS Configuration	Use HTTPS (TLS) (default) Use HTTP (no TLS)
Data Stores	Project Store: C:\ptc\studio-es\project (default) Representations Store: C:\ptc\studio-es\reps (default) Upgrade Store: C:\ptc\studio-es\upgrade (default)
Model Target Generation	<input type="checkbox"/> Enable server-side model target generation
Default Domain Name	The Experience Service needs the 'defaultDomainName' property to be configured to start successfully Default Domain Name: computername.local (example)
Authentication	<input type="checkbox"/> Basic Authentication <input type="checkbox"/> Single Sign-On (OpenID Connect)

From Location	Action
ThingWorx Server	ThingWorx Server Url: http://localhost:8080/Thingworx (example) Configure Public Access to ThingWorx Server Administrator Credentials for ThingWorx Server Username: Administrator Password: *****
Ready to Install	Setup is now ready to begin installing Experience Service on your computer. Restart after completed.

ThingWorx Foundation and Vuforia Experience Services can be installed on separate servers and have separate PostgreSQL servers to improve performance. This document covers a single-server installation. The following capture is an example of a successful connection of Vuforia Experience Services. If the Vuforia Experience Service errors and closes, See [Troubleshooting on page 117](#). Otherwise, minimize this window (do not close the window) when the server is listening.

```

npm
vxs 138|13,21,3,1 options /VuforiaExperienceService/compliance/views/ options
vxs 139|13,21,3,2 /VuforiaExperienceService/compliance/views/ sendMethodNotAllowed
vxs 140|13,21,4 get /VuforiaExperienceService/compliance/users/ <anonymous>
vxs 141|13,21,4,1 options /VuforiaExperienceService/compliance/users/ options
vxs 142|13,21,4,2 /VuforiaExperienceService/compliance/users/ sendMethodNotAllowed +21ms
vxs ## Routes report ends +65ms
Setting DEBUG value: undefined
Enabled debug namespaces:
Disabled debug namespaces: follow-redirects verb.vxs:apid.rt verbose.vxs:acl verbose.vxs:acldbAcldbBackend verbose.vxs:auth.tokenManager verbose.vxs:cds.metadata verbose.vxs:cds.unzip verbose.vxs:irs.search verbose.vxs:pghelper verbose.vxs:proxy.vxs:access.deleteApplListener vxs:acl vxs:acl.TwRolesSync vxs:acl.pkg vxs:acl.roleMembers vxs:acldbAcldbBackend vxs:admin.domain vxs:admin.settings vxs:admin.users vxs:app vxs:auth vxs:auth.tokenManager vxs:auth.tw vxs:billable vxs:cds vxs:cds.delete vxs:cds.download vxs:cds.list vxs:cds.listReps vxs:cds.metadata vxs:cds.options vxs:cds.unzip vxs:cons vxs:cryptoHelper vxs:device vxs:digest vxs:dns vxs:fixtimer vxs:irs.delete vxs:irs.get vxs:irs.get.issues vxs:irs.import vxs:irs.options vxs:irs.search vxs:irs.search vxs:irs.upload vxs:irs.utils vxs:limitsHelper vxs:logs vxs:metrics.usage vxs:mtg vxs:ns vxs:ns.acl vxs:ns.settingsLoader vxs:performance-metrics vxs:pghelper vxs:project vxs:project-access vxs:proxy vxs:proxyMgr vxs:pxapi vxs:rep vxs:rep-access vxs:repViewedEvent vxs:resource-access vxs:roles vxs:roles.mgr vxs:security vxs:sql.pg vxs:sql.scriptfilter vxs:tenants vxs:tg vxs:tg.mm vxs:tg.util vxs:upgrade vxs:upgrade.migrators.mbu vxs:upgrade.migrators.metrics vxs:upgrade.migrators.mpp vxs:upgrade.migrators.ss vxs:upgrade.migrators.tg vxs:usage
Monitor vxs:userRegistration vxs:vdp-model-target
Starting server now ...
1 websocket proxies created
Server is using Tmx-01.local as the defaultDomainName
Server is using the 'all' routingMode
Server startup completed in 6655 ms
HTTP Listening on port 2019

```



The Experience Service start.bat can be found in the C:\ptc\studio-es\bin\ directory. Create a shortcut and paste it to the Windows startup folder to launch and minimize the service after Windows starts.

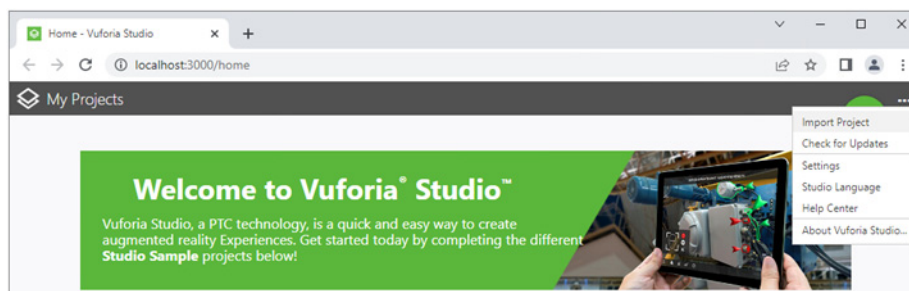
Generic Templates

The Process Library provides several Vuforia generic templates for common process objects to provide users a starting point for designing their own custom experiences. Use the following steps to get started with the Vuforia generic templates.

Import Generic Template

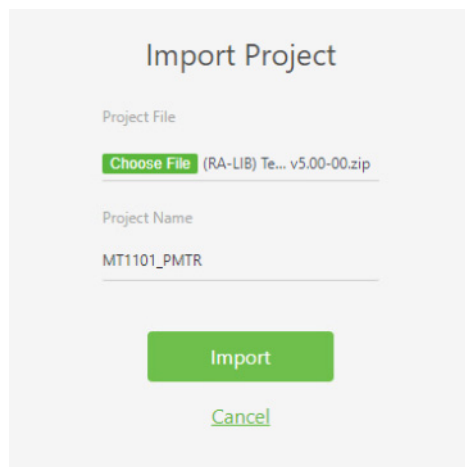
The first step is to import a Vuforia generic template that corresponds to your device. (Example: PAI, PAO, PDO, PDOSE, PMTR, PPID, PVLV, or PVSD)

1. Launch Vuforia Studio and select “Open”.
2. Vuforia Studio opens a page in the default web browser. Select the ellipse on the upper right of the page.

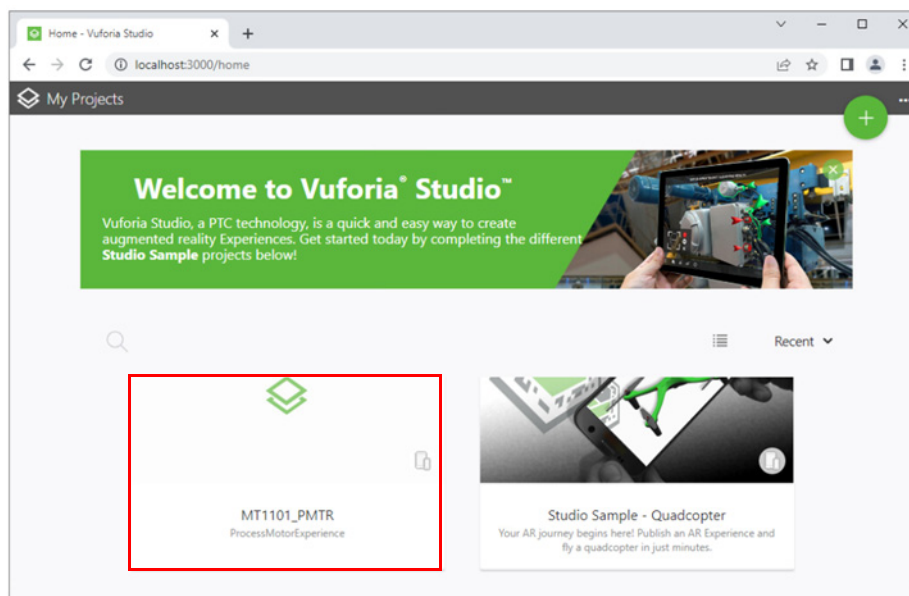


3. Select the “Import Project” button and navigate to the Process Library > Templates > PTC Content > 4. Vuforia Experiences > Vuforia Generic Templates and select a template.

4. Change the "Project Name" to something that identifies your equipment, device, or object and select "Import". This example uses process motor PMTR template.



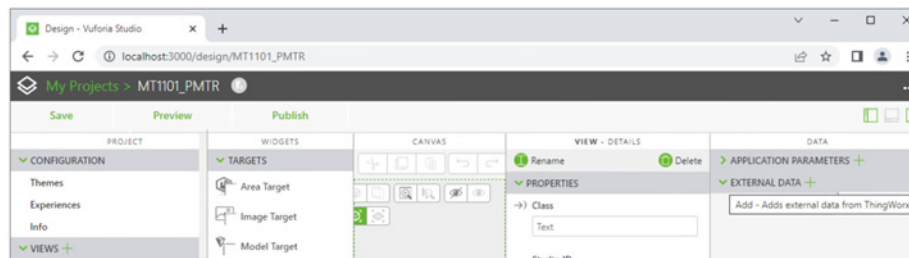
5. After the template has been imported, select the project in Vuforia Studio to open.



Bind External Data

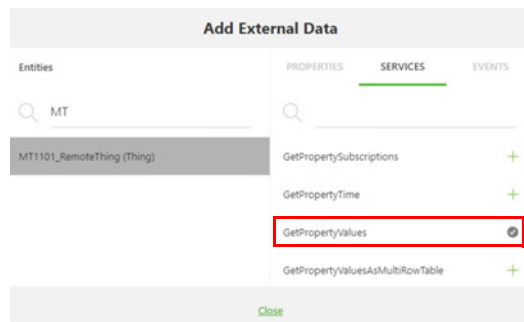
The Vuforia Studio data panel will allow you to bind external data for live-data use in the Vuforia experience. We will use the Control Strategy Health Dashboards process object Things, of type RemoteThings, from ThingWorx Foundation to this project. Follow the steps below to bind your project.

1. With the project opened, select to "Open the Data Panel" using the button in the upper right of the project.

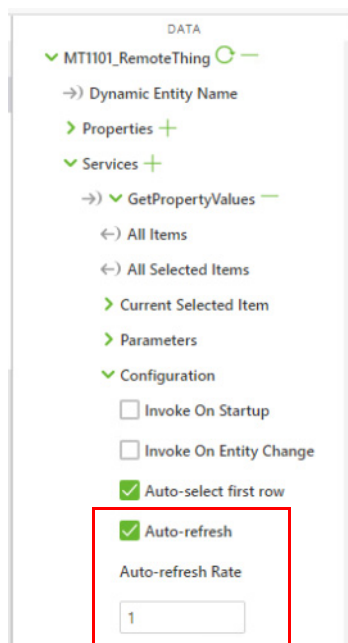


2. Select the - sign under "External Data" to remove any external data connection the template may have referenced prior.
3. Select the + sign to the right of "External Data" to add external data.
4. Browse to find the name of your process object thing. For this example, we select the MT1101_RemoteThing for a process motor PMTR.

- Once highlight, select “Services” menu and scroll down to select “GetPropertyValues”, then click “Close”.



- The project will now contain a binding to your RemoteThing. Verify you can expand and see the “Current Selected Items”.
- Next expand the “Configuration” section and select Auto Refresh. Enter an appropriate scan rate to get the real-time data from the controller.



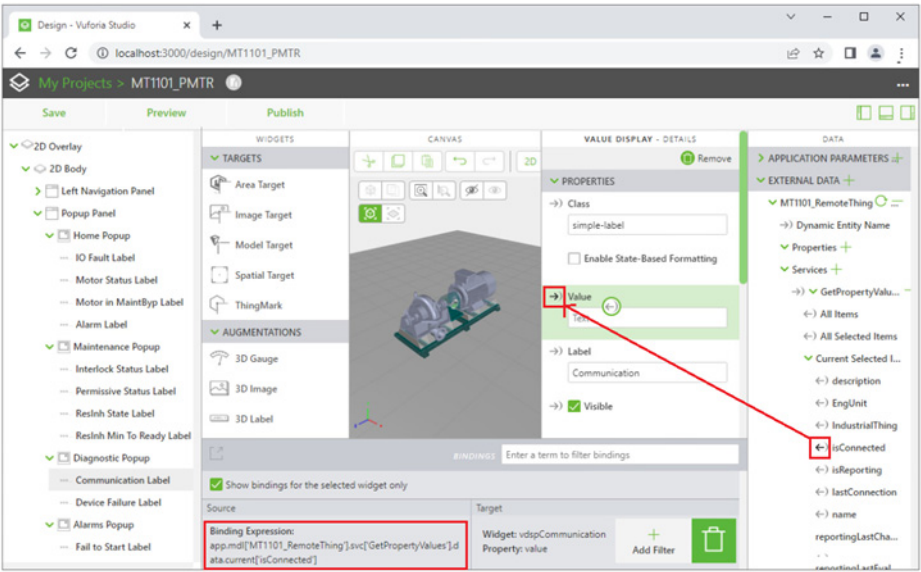
Popup Panel Labels

The PlantPax generic templates provide the PlantPax buttons, graphics, and popup labels for meaningful object parameters. The button and popup content are on the 2D Overlay in the Left Navigation Panel and right Popup Panel.

The Left Navigation Panel is functional without modification. The Popup Panel requires binding of external data items to the “Value” property. Follow the steps before for each popup.

- In the Project panel, select a Popup Label to be configured.
- In the Value Display panel, locate the Value property box. The right pointing arrow ->) is light gray, indicating no binding expression.
- In the External Data panel, browse the current selected items for the RemoteThing tag that corresponds to the Popup. See the table below for item correlation examples.

4. Drag-and-drop the corresponding item left pointing arrow <-) to the corresponding Value property arrow in the Value Display panel.




Once linked, the small arrows display as bold arrows. The bottom Bindings Panel allows you to verify that your bindings have been connected properly.

5. Repeat [step 1](#) ... [step 4](#) for all Popup Labels.

Vuforia generic templates for each process object require binding items from their corresponding external data RemoteThing. The following table provides an example for the PMTR Popup expression binding.

From PMTR Template	From External Data > PMTR RemoteThing > Services > GetPropertyValues > Current Selected Item
Home Popup	<ul style="list-style-type: none">• IO Fault Label - drag "Sts_IQFault" to the value property• Motor Status Label - drag "Sts_eSts" to the value property• Motor in MaintByp Label - drag "Sts_MaintByp" to the Value property• Alarm Label - drag "Sts_Alm" to the Value property
Maintenance Popup	<ul style="list-style-type: none">• Interlock Status Label - drag "Sts_IntlkOK" to the value property• Permissive Status Label - drag "Sts_PermOK" to the value property• Reslnh State Label - drag "Reslnh_StsReady" to the value property• Reslnh Min to Ready Label - drag "Val_MinToReady" to the value property
Diagnostic Popup	<ul style="list-style-type: none">• Communication Label - drag "isConnected" to the value property• Device Failure Label - drag "Sts_NotRdy" to the value property
Alarms Popup	<ul style="list-style-type: none">• Failed to Start Label - drag "Sts_FailToStart" to the value property• Failed to Stop Label - drag "Sts_FailToStop" to the value property• Motor Fault Label - drag "Sts_MotorFault" to the value property
Runtime Popup	<ul style="list-style-type: none">• Current Run Hours Label - drag "Val_CurRunHrs" to the value property• Max Run Hours Label - drag "Val_MaxRunHrs" to the value property• Total Run Time Hours - drag "Val_TotRunHrs" to the value property• Total Number of Starts - drag "Val_Starts" to the value property

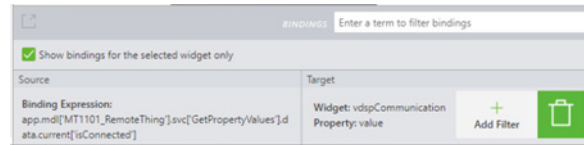
 Users can modify buttons and parameters as needed or add new items. For new items, start by adding the tags to the OPC Aggregator device and then modify or create a RemoteThing using the ThingWorx Composer™. Afterwards, ensure that the External Data binding in Vuforia can browse new items.

Popup Data Filters

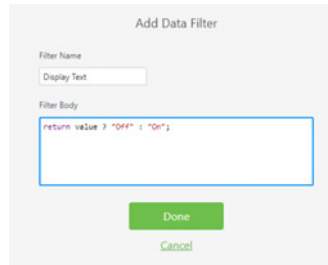
If you want to alter or change the value of the binding before the value is set on the target object, you can add a JavaScript snippet using Add Filter to perform that change.

1. Select the Popup Label you wish to add a data filter.

- From the bottom Binding Panel, click the "Add Filter" button.



- The Add Data Filter window launches. Enter your formula into the filter body for a given item/function.
The following example is a filter of a digital output binding "Sts_Out" so the value of "0" is changed to "Off" and a value of "1" is changed to "On".



The following table contains some JavaScript snippet examples for other item types.

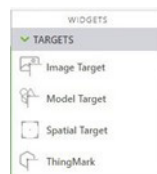
PMTR Items	Filter Body JavaScript snippet
IO Fault "Sts_eSts"	"return value ? "Faulted" : "Not Faulted";
Motor Status "Sts_eSts"	"return value in pMotor.statusEnum ? pMotor.statusEnum [value] : pMotor.statusEnum[0];
Maintenance Byp "Sts_MaintByp"	"return value ? "Bypass" : "Not Bypass";
Alarm Indicator "Sts_Alm"	"return value ? "Not OK" : "OK";
Interlock STS "Sts_IntlkOK"	"return value ? "Failed" : "OK";
ResInh State "ResInh_StsReady"	"return value ? "Ready" : "Not Ready";
ResInh Min to Ready "Val_MinToReady"	"return common.numberFormat(value,2);
Communication "isConnected"	"return value in pMotor.sourceQualityEnum ? pMotor.sourceQualityEnum[value] : pMotor.sourceQualityEnum[0];

3D Model and ThingMark

Replace the 3D drawings included in the template with the associated equipment drawing of your application.

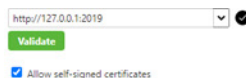
Experiences in Vuforia View are displayed by scanning a ThingMark from your device. For the 3D model to display in the correct location, you must place a digital ThingMark in the same location as your real-world ThingMark.

- Ensure that your Vuforia Studio has licensed Vuforia ThingMarks that can be associated with your equipment.
- Configure the associated ThingMark per experience via the experiences tab under configuration. ThingMarks can be retrieved via Configuration > My ThingMarks.
- Place ThingMark onto the desired location of the piece of equipment within the 3D Canvas. The ThingMark object is available via the Widgets > Targets side menu.



- Place the ThingMark at the same location on your equipment in the field.
- Configure the Experience Information.
- Go to Configuration > Info and enter a description for the Experience. Naming your experience lets the program know which experience to load when a ThingMark is scanned.

7. Enter your Vuforia Experience Service Address and validate via the Validate button.

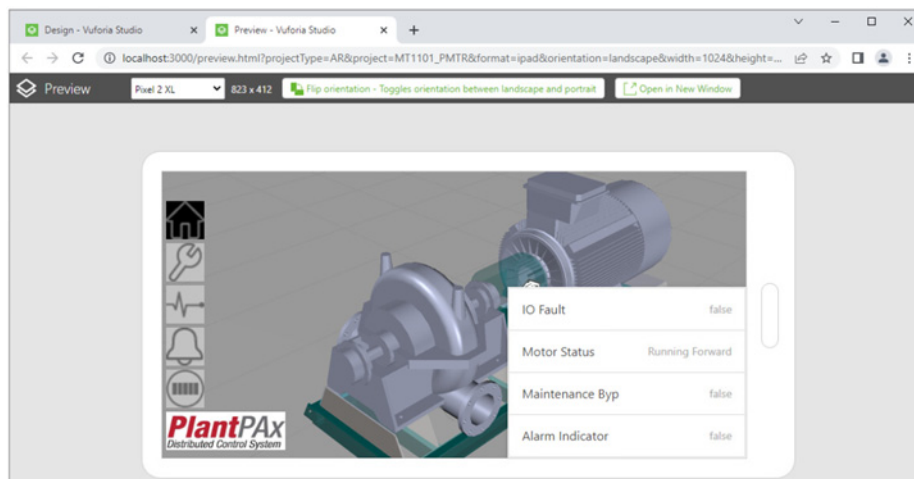


8. Enable Authentication to confirm that appropriate users can access the Experiences.

When all edits are complete, you can preview the Experience (good for testing purposes) or publish the Experience. Once published, you scan the ThingMark associated with the experience to pull up the Augmented Reality Experience.

Preview and Publish

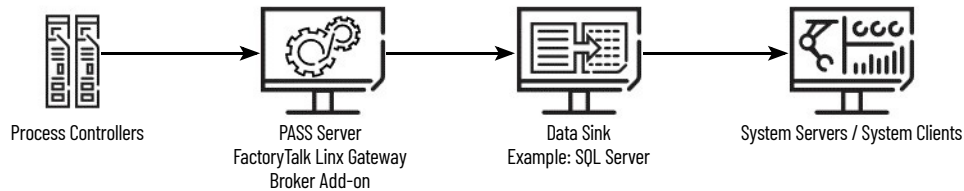
The Vuforia Studio provides Save, Preview, and Publish buttons on the top menu of a project. Select the “Preview” button and another browser tab opens to test your PlantPAx navigation buttons and popups. You can also preview how the experience looks on different mobile device types using the drop-down on the top menu.



Test your project and once it's completed, select the “Publish” button. Now that you've finished, check it out in Vuforia View!

PlantPAx Event Recording - EventQ

About



The Broker-addon PlantPAx® EventQ solution enables the capture of controller-based events to respective data sinks, including SQL database (such as FTAE and BatchHistoryEx), flat files, and so on, for higher-level analytics and process tracking. Each event record is captured locally in the controller and stored in a user configurable journal buffer before unloading data to the respective data sink. Controller event buffer size is determined from the number of event records that are anticipated to occur during an unexpected disconnection from the FactoryTalk® Edge Gateway™ Broker-addon PlantPAx EventQ solution. The Broker-addon PlantPAx EventQ solution significantly improves time stamp resolution of controller-based events. The Broker-addon PlantPAx EventQ solution uses Add-On Instructions in the controller to capture, time stamp and buffer event records. The buffered event records are then formatted and offloaded to selected data sinks using FactoryTalk Edge Gateway.

This chapter covers:

- Configuration of FactoryTalk Edge Gateway and Broker-addon PlantPAx EventQ solution
- Steps to configure pre-defined and user-defined events, mapping of their attributes to a target schema and other miscellaneous configurations.
- Steps to initiate the processor program that captures the event data interfacing with Broker Add-On and persisting it to respective data sinks (SQL, Flat files, and so on).

Prerequisites

Software pre-requisites for the FactoryTalk Edge Gateway EventQ Add-On solution include:

- FactoryTalk Services 6.20.00 (CPR9 SR12) [including FactoryTalk Linx]
- FactoryTalk Edge Gateway 1.03.00
- FactoryTalk Edge Gateway Broker-addon 3.7.1
- Windows® 10
- Windows Server 2016 Standard
- Windows Server 2019 Standard
- Microsoft® SQL Server 2019

Broker Add-On

Broker Add-On is an Add-On application that can be run as part of the FactoryTalk Edge Gateway (FTEG). This Add-On serves as a broker to stream data between controllers and the consuming modules/solutions. The Broker-addon PlantPAx EventQ solution runs as a “solution” as part of the latest version of Broker Add-On.

Known Limitations:

- Length delimiter configuration is supported only for Data Source Driver EtherNet/IP™ (CIP™)
- Pre-defined Record Type LBSM only supports character delimiter configuration.

- UDT String_EventQData maximum characters extension 492. Default 128.

Database Objects

Broker-addon PlantPax EventQ solution captures event data by listening to the Broker-Addon based on the specified configuration and persists the data to different sinks. One of the data sink options that are provided for persisting the event data is a Microsoft SQL Server database. The following are some of the assumptions regarding the setup of Microsoft SQL Server:

1. An instance of Microsoft SQL Server is configured on the Edge/Industrial machine/desktop where FTEG is running or a remote machine/desktop that is accessible to the Edge/Industrial Machine running FTEG over TCP (on port 1433).
2. To adhere to strong security for the Broker-addon PlantPax EventQ solution, SQL Authentication and Integrated Windows Authentication (IWA) is supported. Hence, the Windows Users Contexts under which FTEG is running must have access to the SQL Server. This means the following:
 - a. A login must exist on the SQL Server for the SQL Users and Windows Users (domain\username) who run FTEG
 - b. A user must be created on the database that is associated with the SQL Server login and must have db_owner permissions on the database. This is required since the functionality of Broker-addon PlantPax EventQ solution requires creation of Tables in the database for pre-defined Event Types such as LBSM, RAC_EVENT, and LGXEVENT.
 - c. Grant the following minimal permission for assigned users or roles for the database (Connect, Execute, and Insert).

Note: It is possible for Database Administrators to provide much more granular permissions instead of providing db_owner. However, this topic is out of the scope of the current solution.

Flat File Persistence

For Flat File Persistence of known formats (CSV, XML, JSON, Raw Data) on local disk or an external disk, it is assumed that the Current Windows User running FTEG has access to the specific directories to which they will be persisted.

Controller Add-On Instructions

Add-On Instructions are reusable code objects that contain encapsulated logic that can streamline implementing your system. The Add-On Instructions let you create your own instruction set for programming logic as a supplement to the instruction set provided natively in the ControlLogix® firmware. An Add-On Instruction is defined once in each controller project and can be instantiated multiple times in your application code as needed.

Event Queue Unload (raP_Tec_EventQUnload)

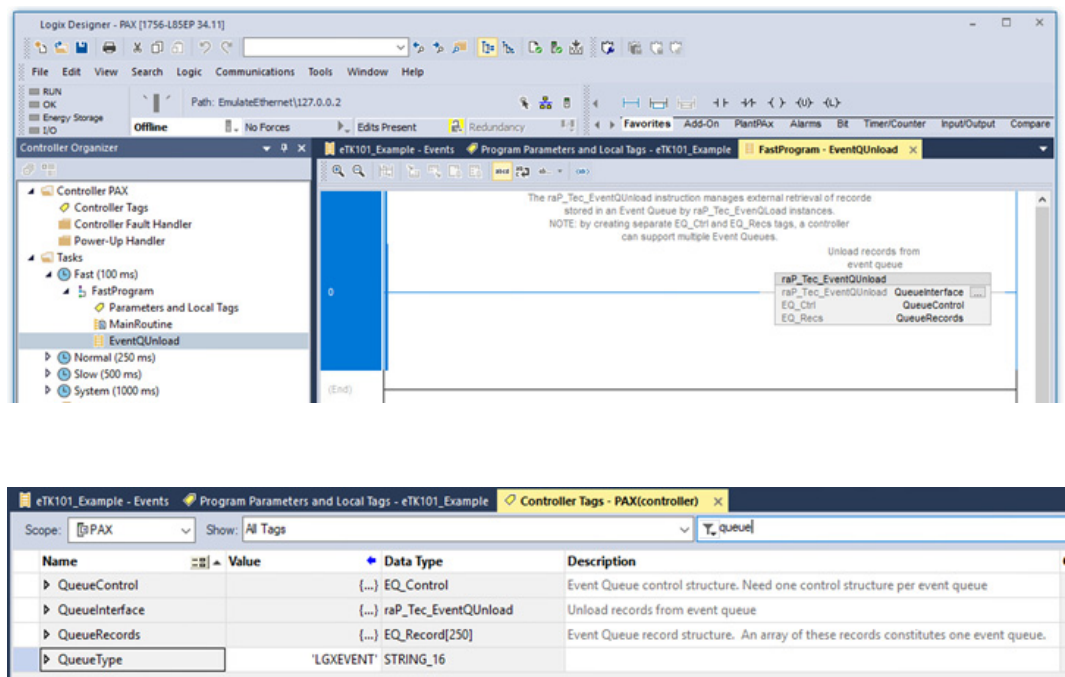
The process library provides an Add-On Instruction for Event Queue Unload. The raP_Tec_EventQUnload instruction manages external access to records and unloads records from the event queue. Use the following steps to import this instruction into your project.

1. Open your controller application using Studio 5000 Logix Designer® and create a routine in which to import the Add-On Instruction.
2. In the routine, right-click and select "Import Rungs". Browse the Process Library > Studio 5000 Logix Designer Files - L5K directory and select the file that is named "raP_Tec_EventQUnload_5.xx.xx.L5X".



Each controller requires at least one "controller-scoped" raP_Tec_EventQUnload instruction. It's recommended to code this instruction in the fastest task available in your application.


The following programming example shows the raP_Tec_EventQUnload instruction.



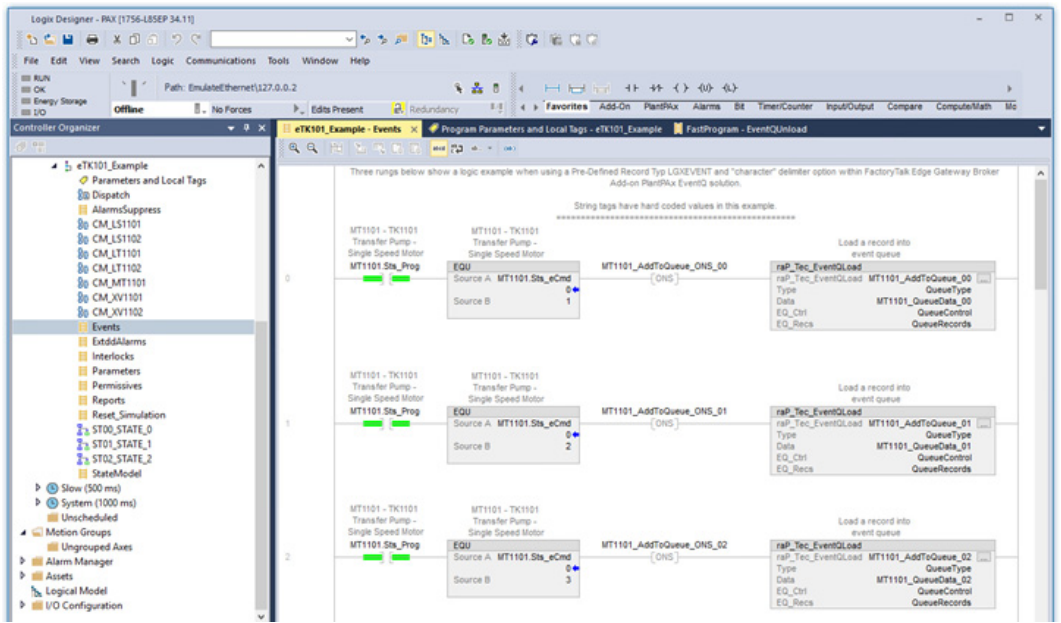
Event Queue Load (raP_Tec_EventQLoad)

The process library provides an Add-On Instruction for Event Queue Load. This raP_Tec_EventQLoad instruction captures and time stamps the event and loads the record into the event queue buffer. Use the following steps to import this instruction into your project

- 1. Open your controller application using Studio 5000 Logix Designer and create a routine in which to import the Add-On Instruction.
- 2. In the routine, right-click and select "Import Rungs". Browse the Process Library > Studio 5000 Logix Designer Files - L5K directory and select the file that is named "raP_Tec_EventQLoad_5.xx.xx.L5X".

 A raP_Tec_EventQLoad instruction is required for each QueueData tag element being logged. The load instructions are typically program-scoped and placed in a routine where the tag element is executed.

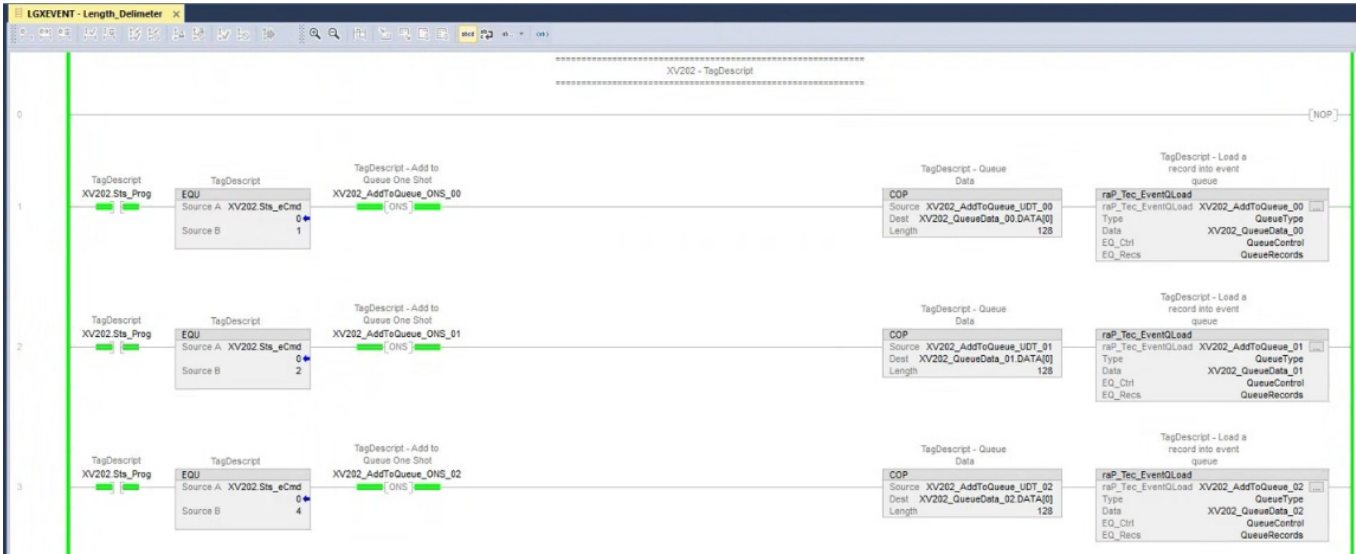
The following example shows example logic when using Pre-Defined Record Type LGXEVENT and character delimiter option within FactoryTalk Edge Gateway Broker Add-On PlantPax EventQ solution. Three raP_Tec_EventQLoad instructions reference QueueData tag elements. Each element of the String_EventData type tag contains a unique string value. The first tag element references a process motor Stop command, the second references the Start command and the third references the Start Reverse command. Any program changes for Stop/Start/StartReverse status will be recorded.



The string data that gets decoded by FactoryTalk Edge Gateway Broker-addon PlantPax EventQ solution can either be “hardcoded” per the following illustration, or the string data can be constructed with logic dynamically.

Name	Usage	Value	Data Type
MT1101.QueueData_00	Local	'PAX,Program:eTK101_Example.MT1101.PCmd_Stop,Program Command to Stop'	String_EventQData
MT1101.QueueData_01	Local	'PAX,Program:eTK101_Example.MT1101.PCmd_Start1,Program Command to Start'	String_EventQData
MT1101.QueueData_02	Local	'PAX,Program:eTK101_Example.MT1101.PCmd_Start2,Program Command to Start Reverse'	String_EventQData

The following example shows example logic when using Pre-Defined Record Type LGXEVENT and “length delimiter” option within FactoryTalk Edge Gateway Broker Add-On PlantPax EventQ solution.



The string data within the user-defined tags can either be “hardcoded” per the following illustration, or the string data can be copied into the user-defined tag structure with logic.

Name	Usage	Value	Force Mask	Style	Data Type	Description	Constant
▲ XV202_AddToQueue_UDT_00	Local	{...}	{...}		raP_UDT_LGXEVENT	TagDescript - User-Defined	<input type="checkbox"/>
▸ XV202_AddToQueue_UDT_00.SourceName		'PPAX'	{...}		STRING_12	TagDescript - User-Defined	
▸ XV202_AddToQueue_UDT_00.TagName		'XV202.PCmd_Pos1'	{...}		STRING_40	TagDescript - User-Defined	
▸ XV202_AddToQueue_UDT_00.Message		'Program Command to Close Len'	{...}		STRING_40	TagDescript - User-Defined	
▲ XV202_AddToQueue_UDT_01	Local	{...}	{...}		raP_UDT_LGXEVENT	TagDescript - User-Defined	<input type="checkbox"/>
▸ XV202_AddToQueue_UDT_01.SourceName		'PPAX'	{...}		STRING_12	TagDescript - User-Defined	
▸ XV202_AddToQueue_UDT_01.TagName		'XV202.PCmd_Pos2'	{...}		STRING_40	TagDescript - User-Defined	
▸ XV202_AddToQueue_UDT_01.Message		'Program Command to Open Len'	{...}		STRING_40	TagDescript - User-Defined	
▲ XV202_AddToQueue_UDT_02	Local	{...}	{...}		raP_UDT_LGXEVENT	TagDescript - User-Defined	<input type="checkbox"/>
▸ XV202_AddToQueue_UDT_02.SourceName		'PPAX'	{...}		STRING_12	TagDescript - User-Defined	
▸ XV202_AddToQueue_UDT_02.TagName		'XV202.PCmd_Stop'	{...}		STRING_40	TagDescript - User-Defined	
▸ XV202_AddToQueue_UDT_02.Message		'Program Command to Stop Len'	{...}		STRING_40	TagDescript - User-Defined	

Note: Length delimiter configuration is supported only for Data Source Driver EtherNet/IP (CIP).

Licensing Requirements

Each instance of data type EQ_Control that is part of a FactoryTalk Edge Gateway model requires 14 FactoryTalk Edge Gateway Licensed Tags.

Controller	Controller Tags part of FTEG Model	Licensed Tags Required per Controller
CLX 1	QueueControl_CLX1	14 tags
CLX 2	QueueControl_CLX2	14 tags
	_LBSMJournalCntl	14 tags
CLX 3	QueueControl_CLX3	14 tags
CLX 4	QueueControl_CLX4	14 tags
CLX 5	QueueControl_CLX5	14 tags
	_LBSMJournalCntl	14 tags
	RAC_EVENT_QueueControl	14 tags
Total License Tags Required		112 FactoryTalk Edge Gateway Licenses

Note: The previous table is only an example where multiple instances of data type EQ_Control are used with different tag names and explain the calculation of licensing requirements. Typically, only one instance of EQ_Control is required per controller, but not limited to one.

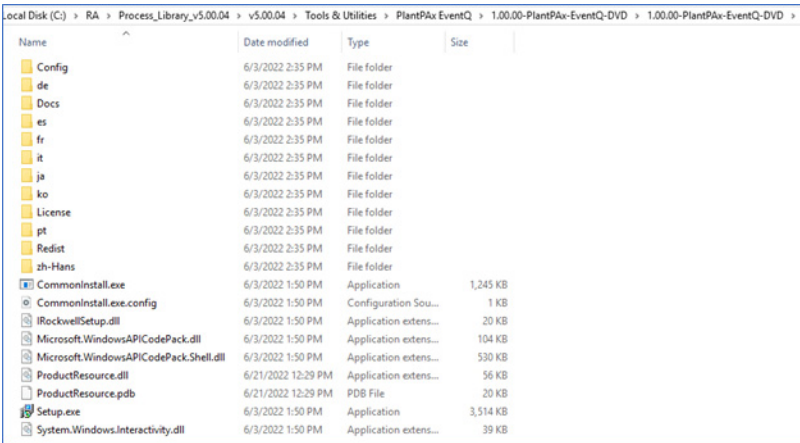
Installing Broker-addon

Controller buffered event records are formatted using FactoryTalk Edge Gateway with a Broker-addon Windows service and offloaded to selected data sinks.

1. Download Library of Process Objects v5.00.04 or later.
2. Navigate to location \Process_Library_v5.xx.xx\Tools & Utilities\PlantPAx EventQ.

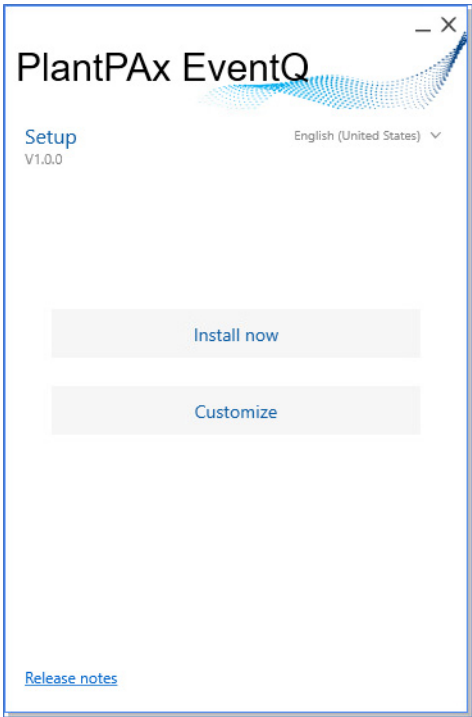
Name	Date modified	Type	Size
1.00.00-PlantPAx-EventQ-DVD.zip	6/22/2022 6:39 AM	Compressed (zipp...	78,664 KB

3. Extract file 1.00.00-PlantPAx-EventQ-DVD.zip.



Name	Date modified	Type	Size
Config	6/3/2022 2:35 PM	File folder	
de	6/3/2022 2:35 PM	File folder	
Docs	6/3/2022 2:35 PM	File folder	
es	6/3/2022 2:35 PM	File folder	
fr	6/3/2022 2:35 PM	File folder	
it	6/3/2022 2:35 PM	File folder	
ja	6/3/2022 2:35 PM	File folder	
ko	6/3/2022 2:35 PM	File folder	
License	6/3/2022 2:35 PM	File folder	
pt	6/3/2022 2:35 PM	File folder	
Redist	6/3/2022 2:35 PM	File folder	
zh-Hans	6/3/2022 2:35 PM	File folder	
CommonInstall.exe	6/3/2022 1:50 PM	Application	1,245 KB
CommonInstall.exe.config	6/3/2022 1:50 PM	Configuration Sou...	1 KB
iRockwellSetup.dll	6/3/2022 1:50 PM	Application extens...	20 KB
Microsoft.WindowsAPICodePack.dll	6/3/2022 1:50 PM	Application extens...	104 KB
Microsoft.WindowsAPICodePack.Shell.dll	6/3/2022 1:50 PM	Application extens...	530 KB
ProductResource.dll	6/21/2022 12:29 PM	Application extens...	56 KB
ProductResource.pdb	6/21/2022 12:29 PM	PDB File	20 KB
Setup.exe	6/3/2022 1:50 PM	Application	3,514 KB
System.Windows.Interactivity.dll	6/3/2022 1:50 PM	Application extens...	39 KB

4. Run Setup.exe as administrator.
5. Select the Install now button. This installs Broker-addon v3.7.1 and PlantPAx EventQ v1.0.0



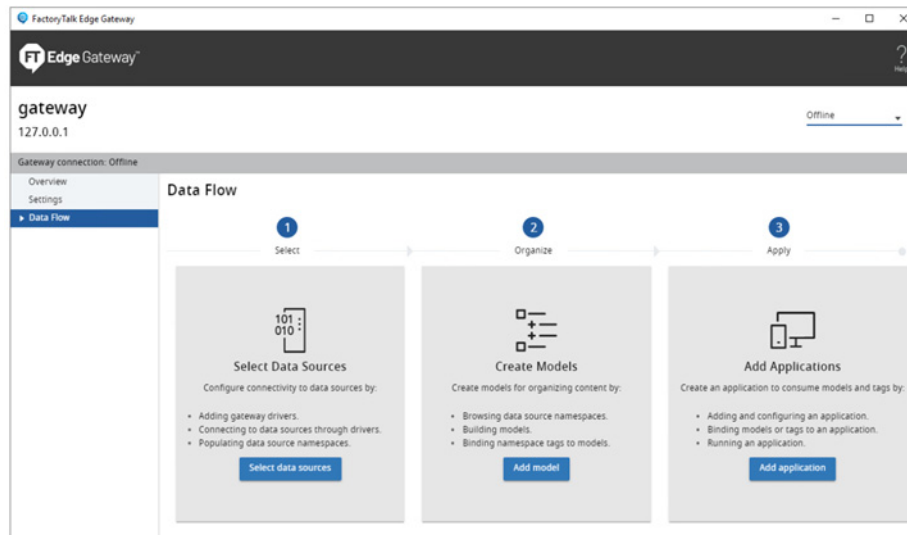
6. A restart is required. When prompted, click restart now.

FactoryTalk Edge Gateway (Data Source)

FactoryTalk Edge Gateway software simplifies and automates collection, contextualization, and organization of industrial equipment data across machines, devices, and automation assets at the source itself—enabling high data integrity from the outset.

Configuration of a FactoryTalk Edge Gateway project is handled in three steps.

- Select Data Sources
- Create Models
- Add Applications



Configure Drivers

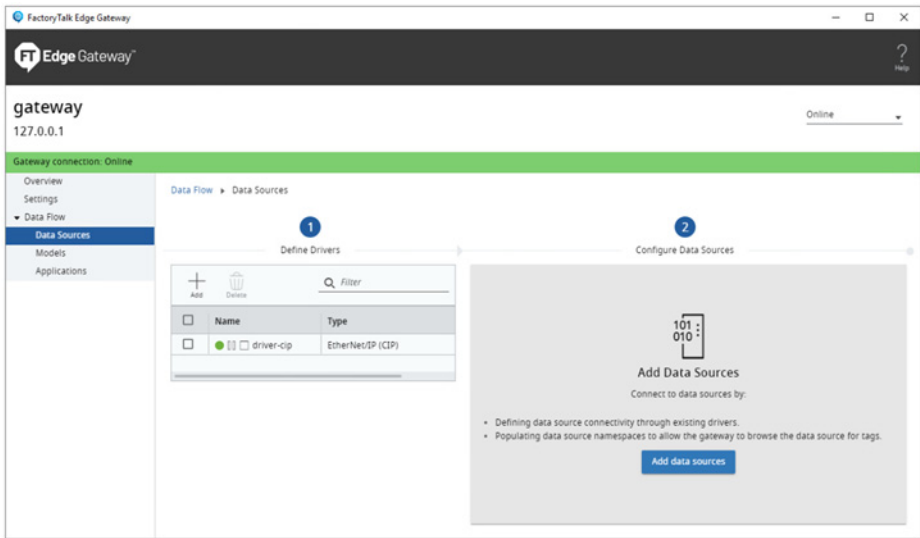
Drivers enable communication between the gateway and data sources. There are two driver options when using the EventQ solution.

- Ethernet/IP (CIP) – a direct connection to a device.
- FactoryTalk® Live Data (FTLD) – a system level connection to a device

The following section describes configuring an existing driver or adding a new driver to FactoryTalk Edge Gateway. Decide which driver best fits your application, then follow the corresponding instructions.

1. Use the Windows start menu to launch the FactoryTalk Edge Gateway Configuration Tool.
2. Go online with FactoryTalk Edge Gateway to monitor the running gateway or modify the current gateway configuration using the offline/online drop-down in the to-right corner.
3. If a configuration mismatch occurs, do one of these actions to go online:
 - Select Upload to restore the previous gateway configuration. Offline modifications are lost.
 - Select Download to overwrite the running gateway configuration with the current modifications. If not backed up, the previous gateway configuration is lost.

4. In the top-left corner, expand “Data Flow” and select “Data Sources”.

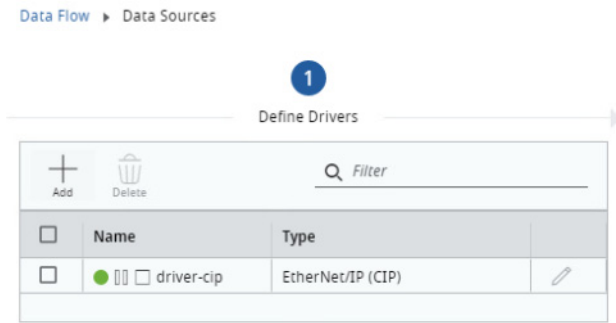


Note: FactoryTalk Edge Gateway by default has an EtherNet/IP (CIP) driver that is added when launched for the first time.

Add Ethernet/IP Driver

Use this option if you want to use the Ethernet/IP (CIP) driver for FactoryTalk Edge Gateway.

- If you want to use the existing Ethernet/IP (CIP) driver, select the Edit icon (pencil).
- If you want to add a new Ethernet/IP (CIP) driver, select the Add icon (plus icon).



1. When adding a new Ethernet/IP (CIP) driver, the select the Ethernet/IP (CIP) type.



If the computer where CIP adapter is running has multiple network interfaces (multiple IP addresses), confirm that the Driver IP address and Driver Default Gateway IP address are configured correctly. That is, the CIP adapter resides on the same network as the target CIP devices. For example: if your computer has two network interfaces that are assigned with the IP addresses (172.18.1.20 & 192.168.1.10) and you want to communicate with CIP device available at 192.168.1.20, then the Driver IP address would be set to 192.168.1.10.

- 2. Enter the machine IP address in Driver IP Address field.

Edit EtherNet/IP (CIP) Driver

Driver Name

driver-cip

Driver IP Address

172.18.1.20

0.0.0.0 will automatically be replaced with the real ip address of the local computer

Driver Default Gateway IP Address

ADVANCED

Save

Cancel

- 3. Start EtherNet/IP (CIP) driver by clicking the play button (triangle icon) in the name column. The status icon turns to a green circle when running.

Data Flow ▶ Data Sources

1

Define Drivers

+

Add

🗑

Delete

🔍

Filter

<input type="checkbox"/>	Name	Type	
<input type="checkbox"/>	<div><div>●</div><div>▶▶▶</div><div>driver-cip</div></div>	EtherNet/IP (CIP)	<div>✎</div>

Add FactoryTalk Live Data Driver

FactoryTalk Live Data (FTLD) is the recommended driver for use with the EventQ solution. The FactoryTalk Live Data (FTLD) driver enables efficient communication between FactoryTalk Edge Gateway and existing FactoryTalk Live Data sources.

- 1. Add FactoryTalk Live Data (FTLD) driver by clicking Add driver button driver (plus icon).

Data Flow ▶ Data Sources

1

Define Drivers

+

Add

🗑

Delete

🔍

Filter

<input type="checkbox"/>	Name	Type	
<input type="checkbox"/>	<div><div>●</div><div>▶▶▶</div><div>driver-cip</div></div>	EtherNet/IP (CIP)	<div>✎</div>

2. Select FactoryTalk Live Data (FTLD) Driver.

Select Driver

☒ FactoryTalk Live Data (FTLD)

☐ EtherNet/IP (CIP)

☐ OPC Data Access (DA)

☐ KEPServer Enterprise (OPC DA)

Next

Cancel

3. Select Login method, enter Username and Password where required and click Save.

Add FactoryTalk Live Data (FTLD) Driver

Driver Name

driver-ftld

☐ Login as Currently SignedIn User

Username

Password

☐ Login to Local Directory

☐ Login to Network Directory

Save

Cancel

4. Confirm that Driver is Running. The status icon turns green when running.

Data Flow ▶ Data Sources

1

Define Drivers

+

Add

🗑

Delete

🔍

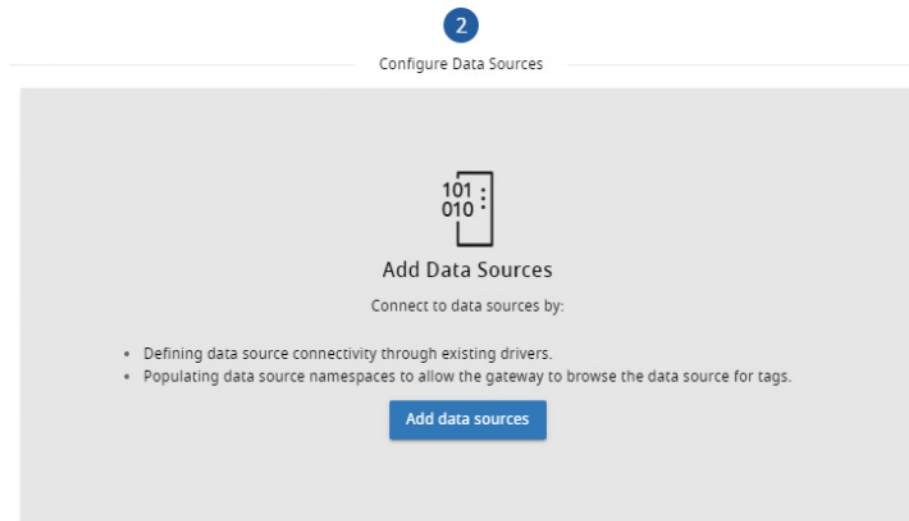
Filter

<input type="checkbox"/>	Name	Type	
<input type="checkbox"/>	<div><div>●</div><div>🔍</div><div>driver-cip</div></div>	EtherNet/IP (CIP)	<div>✎</div>
<input type="checkbox"/>	<div><div>●</div><div>🔍</div><div>driver-ftld</div></div>	FactoryTalk Live Data (FTLD)	<div>✎</div>

Add Data Source

FactoryTalk Edge Gateway gathers the buffer data from the source and stores the data into a namespace. Configure data sources to populate a namespace and collect data from data tags.

1. Select the "Add Data Sources" button.



2. Enter a Data Source Name.



Name the data source using the associated controller name or shortcut topic for easy identification.

3. Select Tags and types defined in (data browse path type) drop-down arrow, and choose Logix Project File (ACD).

4. Browse for Logix Project File (ACD) by clicking Select File and navigate to your Studio 5000 Logix Designer project file.

Configure Data Source

Name
pax

Tags and types defined in (data browse path type)
Logix Project File (ACD)

File path
Select File PAX.ACD

☒ Auto-populate namespace

Driver for collection
Select EtherNet/IP(CIP) driver for Smart Object data collection

Save Cancel

5. Select your driver of choice (Ethernet/IP or FactoryTalk Live Data) from Driver for collection drop-down list.
6. If using the Ethernet/IP (CIP) driver, enter the Collection path.

Configure Data Source

Name
ppax

Tags and types defined in (data browse path type)
Logix Project File (ACD)

File path
Select File PlantPAx_EventQ.ACD

☒ Auto-populate namespace

Driver for collection
driver-cip
Select EtherNet/IP(CIP) driver for Smart Object data collection

Collection path
172.18.1.26:1/4

Save Cancel

7. If using the FactoryTalk Live Data (FTLD) driver, browse the network/local paths to your controller topic.

Configure Data Source

Select File

☒ Auto-populate namespace

Driver for collection
driver-ftld

Select EtherNet/IP(CIP) driver for Smart Object data collection

Collection path
RNA://\$Global/PlantPax/PlantPax_Data_Area::[PAX]

Select Path

▼ Network

▶ FTViewDemo

▶ InstantFizz

▼ PlantPax

▼ PlantPax_Data_Area

PAX

Save

Cancel



The FactoryTalk Edge Gateway User Manual, publication [95055-UM006](#), contains more information on the syntax for collection paths.

8. Initially the data source status displays Loading, then changes to Populated when loading is complete. Confirm that the Collection Path is running (green dot) before continuing.

2

Configure Data Sources

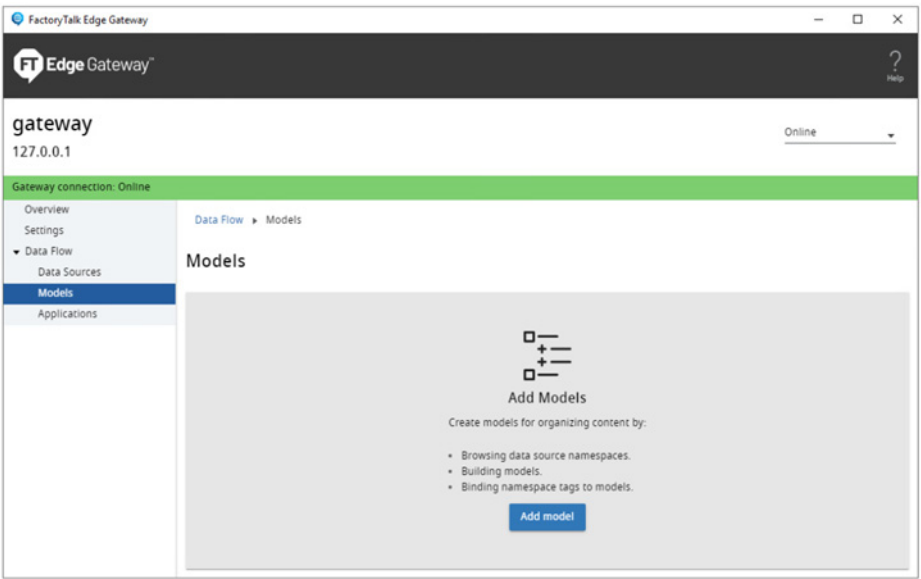
				Filter		
<input type="checkbox"/>	Name	Data Source Type	Status	Driver	Collection Path	
<input type="checkbox"/>	pax	Logix Project File (ACD)	Populated	driver-ftld	RNA://\$Gl...	

FactoryTalk Edge Gateway (Model)

Use models to organize data into a hierarchy. Associate models with applications or data destinations. Build models by staging tags, and then associating the tags to a node or property.

Use the following steps to create a model, stage the 14 attributes from the controller tag of type EQ_Control and bind the tags to the model.

1. Select Data Flow, expand Data Flow and select Models. Create a model by selecting the “Add model” button.



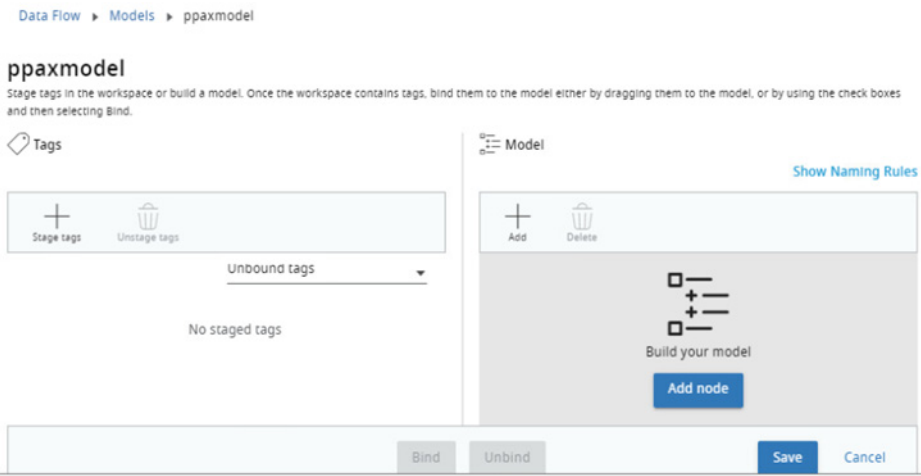
2. Enter a model name and click Save. Note: Configure model name that is associated to site \ area name.



3. Edit model by selecting the edit button (pencil icon).



4. Add node to model by selecting the add node button (plus icon).



5. Enter a node name. As recommended with the data source, name the node so that it contains the controller name or shortcut topic.

Data Flow ▶ Models ▶ ppaxmodel

ppaxmodel

Stage tags in the workspace or build a model. Once the workspace contains tags, bind them to the model either by dragging them to the model, or by using the check boxes and then selecting Bind.

Tags

+

Stage tags

🗑️

Unstage tags

Unbound tags

No staged tags

Model

+

Add

🗑️

Delete

pax_node

Show Naming Rules

Bind

Unbind

Save

Cancel

6. Select the Stage tags button (plus icon).

Data Flow ▶ Models ▶ ppaxmodel

ppaxmodel

Stage tags in the workspace or build a model. Once the workspace contains tags, bind them to the model either by dragging them to the model, or by using the check boxes and then selecting Bind.

Tags

+

Stage tags

🗑️

Unstage tags

Unbound tags

No staged tags

Model

+

Add

🗑️

Delete

pax_node

Show Naming Rules

Bind

Unbind

Save

Cancel

7. Expand your data source and browse for the Queue Control tag (type EQ_Control) used with the raP_Tec_EventQUnload Add-On Instruction.

Select Tags

Data sources ▼ Types ▼ 0 selected

Q QueueControl (1/2416) x

Name	Path	Type
▲ pax		acd
▲ QueueControl	pax/QueueControl	EQ_Control

8. Expand the Queue Control tag (type EQ_Control) and select the following 14 tag members.

EQ_Control Tag Member	Path	Type
.Inp_ConfirmDINT	DataSource/QueueControlTag.Inp_ConfirmDINT	dint
.Sts_ConfirmDINT_Echo	DataSource/QueueControlTag.Sts_ConfirmDINT_Echo	dint
.Sts_Full	DataSource/QueueControlTag.Sts_Full	bool
.Sts_ReadyBOOL	DataSource/QueueControlTag.Sts_ReadyBOOL	bool
.Sts_ReadyDINT	DataSource/QueueControlTag.Sts_ReadyDINT	dint
.Record.Data	DataSource/QueueControlTag.Record.Data	string
.Record.Type	DataSource/QueueControlTag.Record.Type	string
.Record.Time[0]	DataSource/QueueControlTag.Record.Time[0]	dint
.Record.Time[1]	DataSource/QueueControlTag.Record.Time[1]	dint
.Record.Time[2]	DataSource/QueueControlTag.Record.Time[2]	dint
.Record.Time[3]	DataSource/QueueControlTag.Record.Time[3]	dint

EQ_Control Tag Member	Path	Type
.Record.Time[4]	DataSource/QueueControlTag.Record.Time[4]	dint
.Record.Time[5]	DataSource/QueueControlTag.Record.Time[5]	dint
.Record.Time[6]	DataSource/QueueControlTag.Record.Time[6]	dint

Select Tags

Data sources ▾ Types ▾ 14 selected

Q QueueControl (1/2416) x

Name	Path	Type
<input checked="" type="checkbox"/> [4]	pax/QueueControl.Record.Time[4]	dint
<input checked="" type="checkbox"/> [5]	pax/QueueControl.Record.Time[5]	dint
<input checked="" type="checkbox"/> [6]	pax/QueueControl.Record.Time[6]	dint
<input checked="" type="checkbox"/> Type	pax/QueueControl.Record.Type	string
▸ RecsOvf	pax/QueueControl.RecsOvf	EQ_Record[6]
<input type="checkbox"/> Sts_CfgSts	pax/QueueControl.Sts_CfgSts	dint
<input checked="" type="checkbox"/> Sts_ConfirmDINT_Echo	pax/QueueControl.Sts_ConfirmDINT_Echo	dint
<input type="checkbox"/> Sts_Empty	pax/QueueControl.Sts_Empty	bool
<input checked="" type="checkbox"/> Sts_Full	pax/QueueControl.Sts_Full	bool
<input type="checkbox"/> Sts_FullOvf	pax/QueueControl.Sts_FullOvf	bool
<input type="checkbox"/> Sts_GettingFull	pax/QueueControl.Sts_GettingFull	bool
<input checked="" type="checkbox"/> Sts_ReadyBOOL	pax/QueueControl.Sts_ReadyBOOL	bool
<input checked="" type="checkbox"/> Sts_ReadyDINT	pax/QueueControl.Sts_ReadyDINT	dint

Save Cancel

- When the tag browser confirms "14 selected" in the upper right, select the Save button.
- Check each of the 14 unbound tags on left and check node on right.

Data Flow ► Models ► ppaxmodel

ppaxmodel

Stage tags in the workspace or build a model. Once the workspace contains tags, bind them to the model either by dragging them to the model, or by using the check boxes and then selecting Bind.

Tags

Stage tags Unstage tags

Unbound tags ▾

- ☒ pax/QueueControl.Sts_ReadyBOOL
- ☒ pax/QueueControl.Sts_Full
- ☒ pax/QueueControl.Sts_ReadyDINT

Model

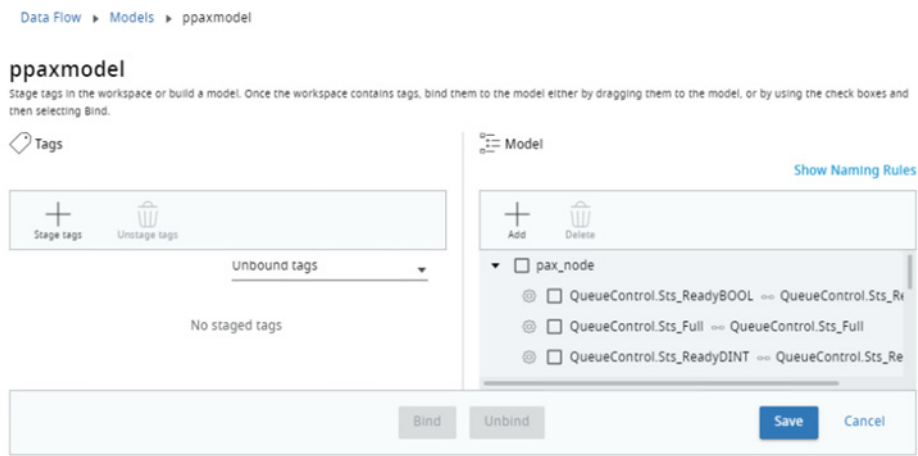
Show Naming Rules

Add Delete

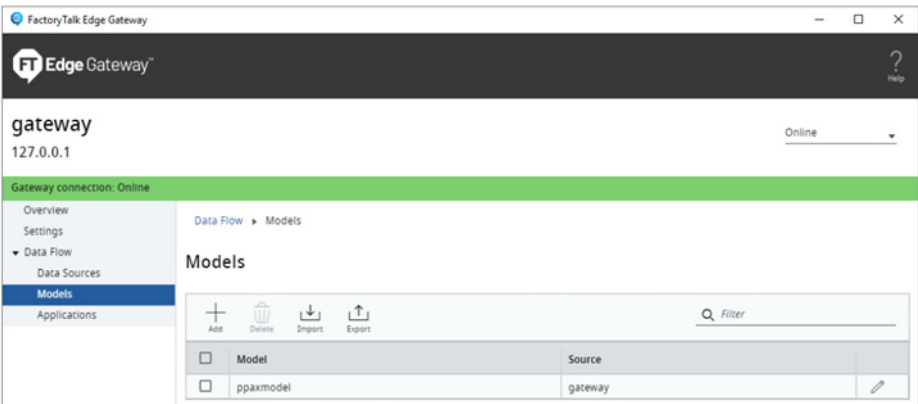
- ☒ pax_node

Bind Unbind Save Cancel

11. The Bind button becomes available. Select Bind and then select Save.



Example of the Data Flow Model Configured

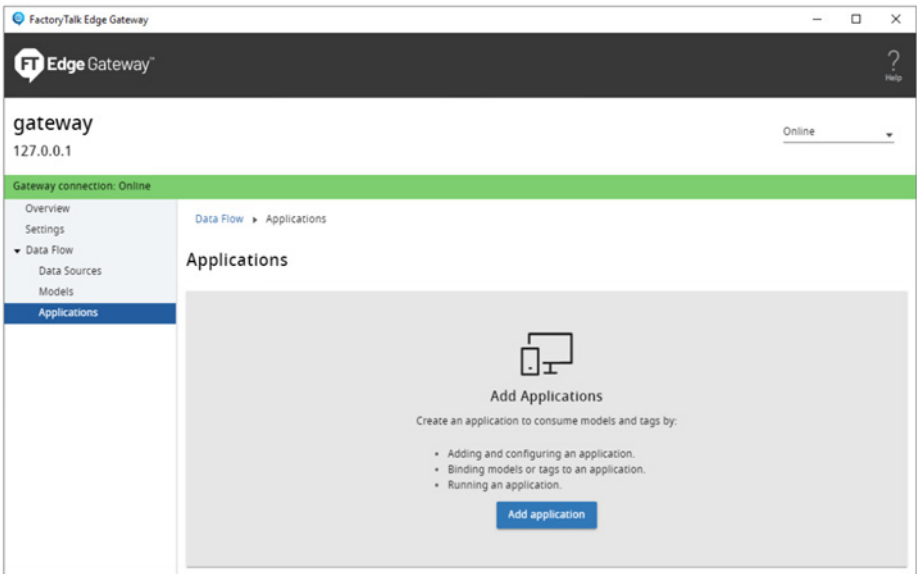


12. Repeat [step 1](#) ... [step 11](#) to create additional model nodes for each controller in your system that uses the EventQ solution and Bind the 14 tags from respective controllers to respective nodes.

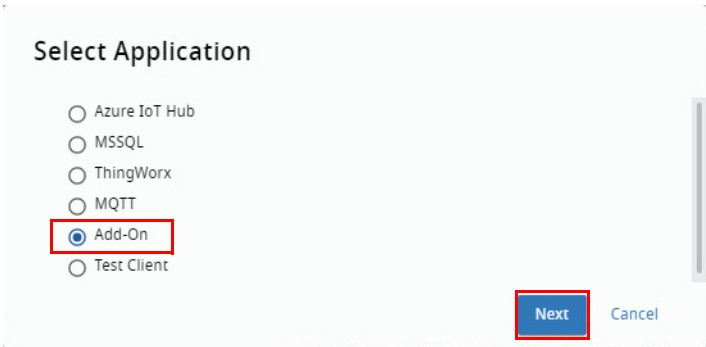
FactoryTalk Edge Gateway (Application)

Applications in FactoryTalk Edge Gateway collect data from tags, models, and then send the data to the configured server.

1. Expand Data Flow and select Applications.

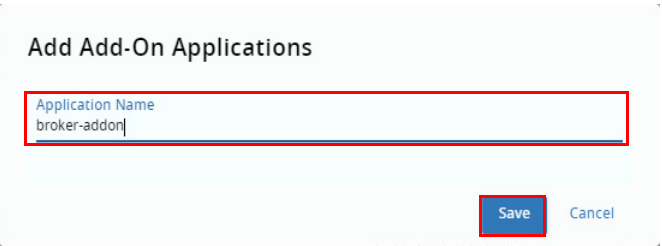
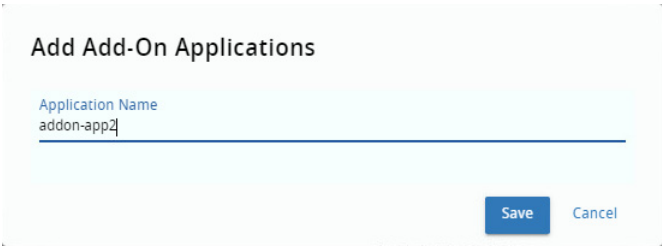


2. Add an application by selecting the Add application button.
3. Select the application type of "Add-On."



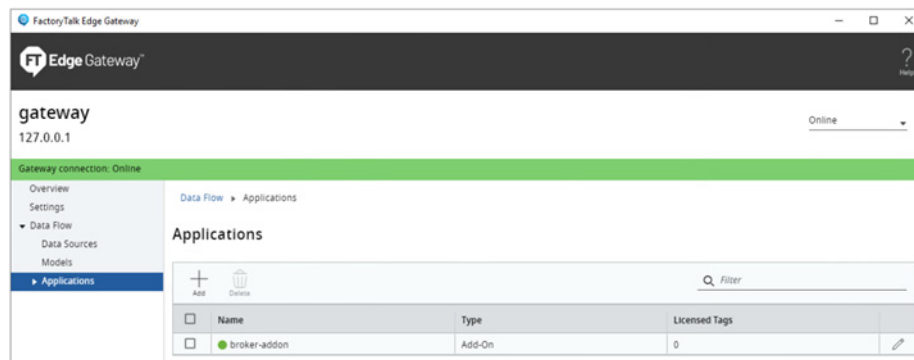
The broker-addon maps event data in a specific format, which is also backward compatible with the prior Logix Batch SequenceManager™ (LBSM) EventQ processor solution.

4. Rename Add-On Applications Name to "broker-addon" and select Save.

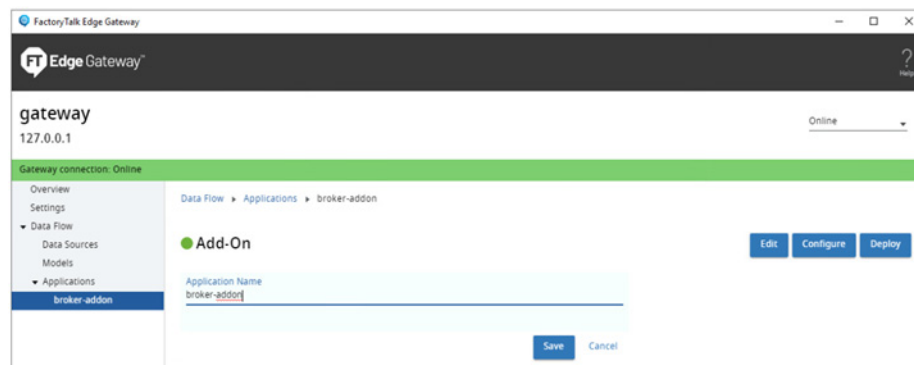


IMPORTANT Application Name must be "broker-addon".

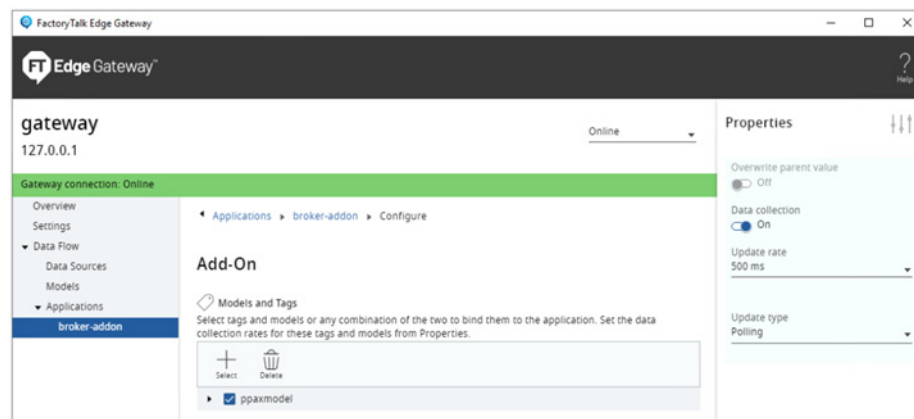
5. Edit the broker-addon application by selecting the edit button (pencil icon)



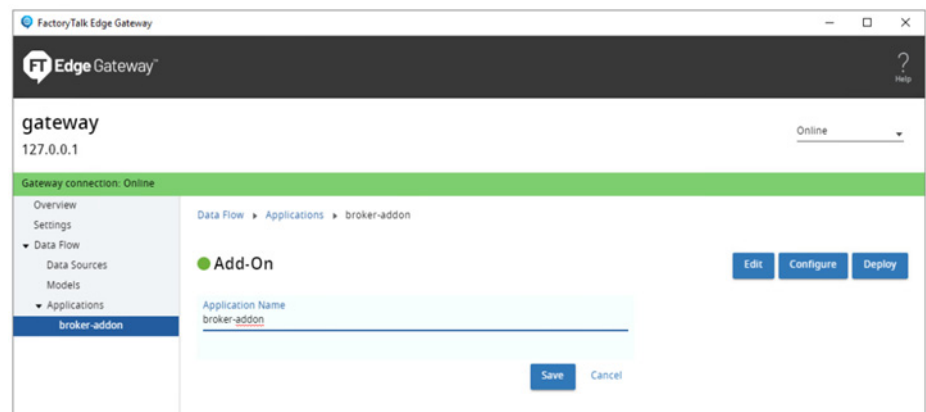
6. Configure the broker-addon application by selecting the Configure button.



7. Under the broker-addon application section, use the Select button or (plus icon) to launch the tag browser.
8. Choose your model by checking the box for specified model and select Save.
9. Select the model and configure the properties on the right windowpane.
 - Data Collection = On
 - Update rate = 500 ms
 - Update type = Polling



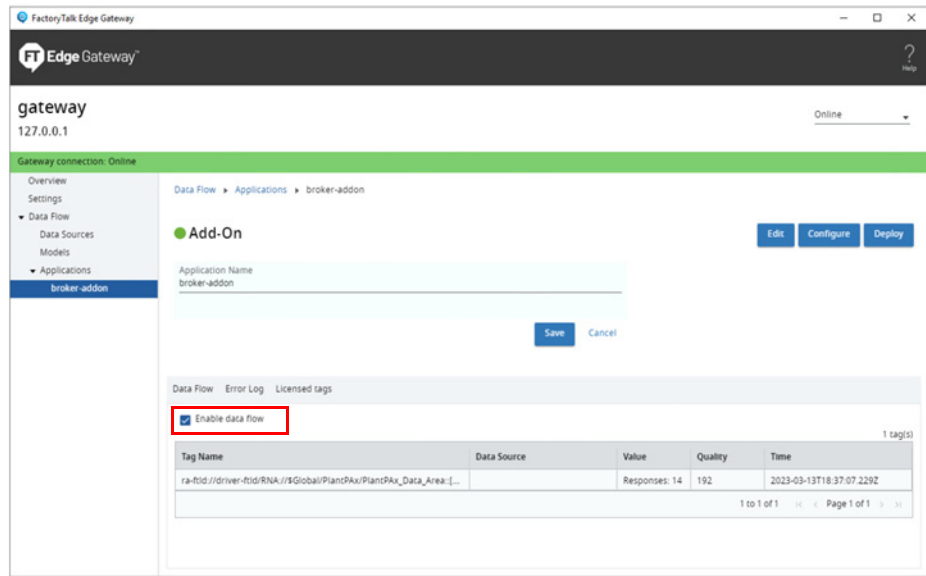
10. Deploy Add-On Application by selecting the Deploy button.



11. Confirm deploy by selecting the Deploy button.



12. Select the Enable data flow checkbox to test successful deployment.



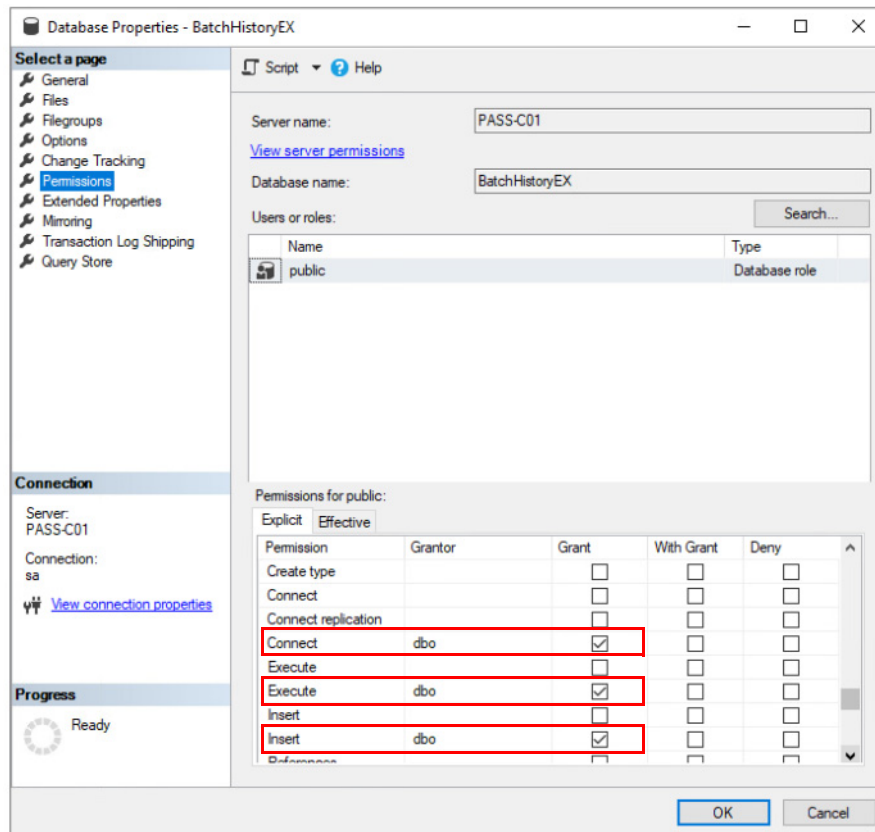
SQL Database Permissions

The broker-addon runs under the NT AUTHORITY\Network Service account for Windows Authentication. Windows Authentication will not work if SQL database is on another machine due to Broker-addon running as a service with Network Service account.

Ensure that your users have sufficient EventQ database permissions. Use the following steps to grant permissions.

1. Open SQL Server Management Studio.
2. In the Object Explorer, expand the Databases folder and find or create the database that you wish to use for EventQ data. (Example: BatchHistoryEX)
3. Select the database, right-click, and select Properties. Select the permissions page.
4. Ensure your authorized users have Connect, Execute, and Insert permissions.

The following example provides permissions for public access to the EventQ database.



Broker-addon (EventQ Solution)

About

Using the Broker-addon PlantPAx EventQ Solution Configuration UI, users can do the following:

- Add, Edit, Copy, and Delete Pre-Defined and User-Defined Record Types.
- Auto populate the information table for pre-defined Record Types. Supported Pre-Defined Record Types:
 - LBSM
 - RAC_EVENT
 - LGXEVENT
- Supported delimiter types:
 - Character

- Length (Length Delimiter is supported only for Data Source Driver EtherNet/IP (CIP).
- Define the output configurations for each record type. Supported output configuration:
 - SQL (Insertion) to SQL Server database
 - CSV
 - JSON
 - XML
 - RAW Data
- Define the SQL columns mapping to the processed record fields and expression. Supported expressions
 - Date
 - Time
 - MilliTIme
 - RecordType
 - GUID
 - TickTimeStamp
 - EventTimeStamp
- Select the date and time format using miscellaneous configuration:

Date format:

 - MMDDYYYY
 - DDMMYYYY
 - YYYYDDMM
 - YYYYMMDD
 - None

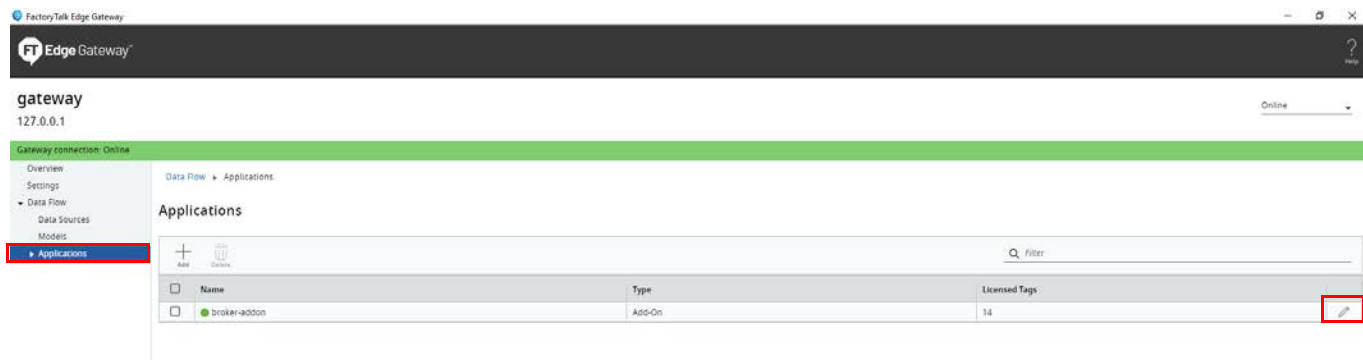
Time format:

 - 24 Hour
 - 12 Hour
 - None

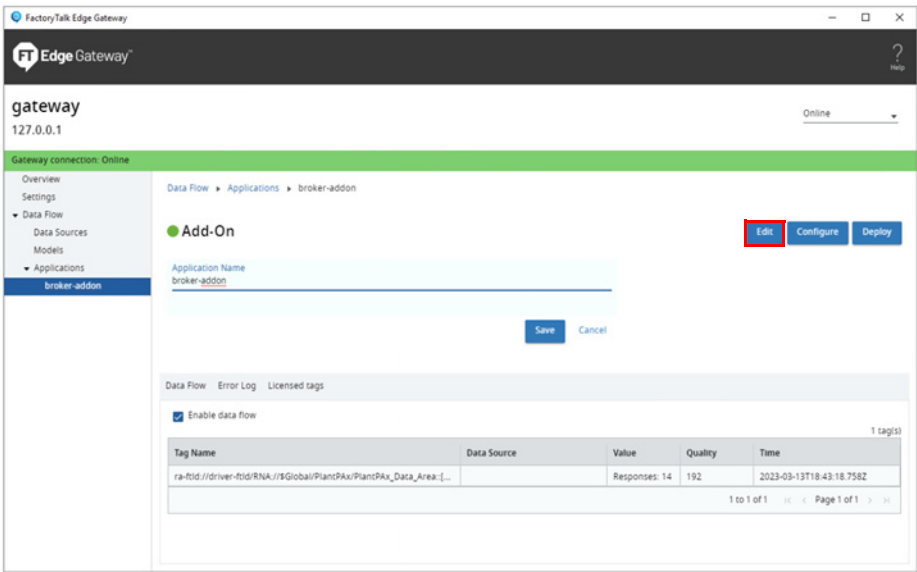
Add PlantPax EventQ Solution

Follow the steps below to add and configure the PlantPax EventQ solution to use the LGXEVENT record type.

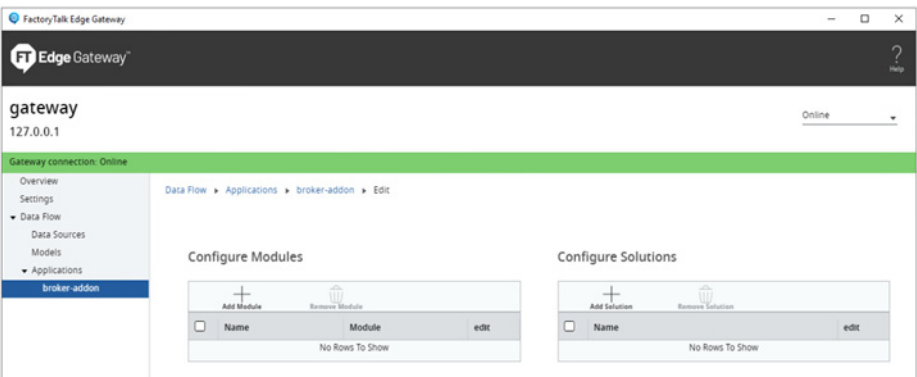
1. Select Data Flow and expand selection.
2. Select Applications and click the edit symbol (pencil icon).



3. Ensure that the Add-On is running and select the edit button.



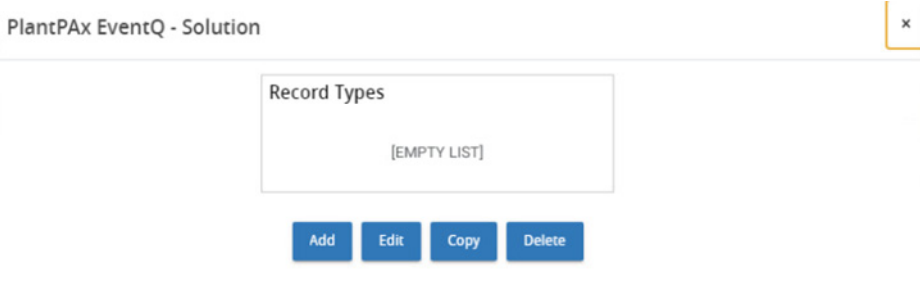
4. Select the Add Solutions button.



5. A popup launches, then select the PlantPAx EventQ solution.



6. The PlantPAx EventQ solution configuration will load and launch another popup where the user can Add \ Edit \ Copy \ Delete Record Types.



When creating a Record Type, the Record Type Name must match the string value that is configured in raP_Tec_EventQLoad attribute Type. Broker-addon PlantPAx EventQ solution determines which mapping schemas to use by comparing the Type string to the Record Type Name. Example, QueueType has a string value of LGXEVENT, thus Record Type Name should be LGXEVENT.

TagDescript - Load a
record into event
queue

raP_Tec_EventQLoad

raP_Tec_EventQLoad

XV101_AddToQueue_00

Type

QueueType

Data

XV101_QueueData_00

EQ_Ctrl

QueueControl

EQ_Recs

QueueRecords

Name	Usage	Value	Force Mask	Style	Data Type	Description	Constant
QueueType	Local	'LGXEVENT'		{...}	STRING_16	Record Type Name	<input type="checkbox"/>

Add Pre-Defined Record Type (Default)

Use the following steps if the predefined structure table values meet your requirements. This document uses the LGXEVENT record type as an example.

- 1. Select the Add button to add a new Record Type. A popup shall appear, provide the name of the Record Type, and select the Add button.

PlantPAx EventQ - Solution

Record Types

Record Type Name

Ex: LGXM, LGXEVENT, RAC_EVENT etc.

Add

Cancel

PlantPAx EventQ - Solution

Record Types

Record Type Name

LGXEVENT

Add

Cancel

- 2. The solution configuration window contains a series of numbered steps.

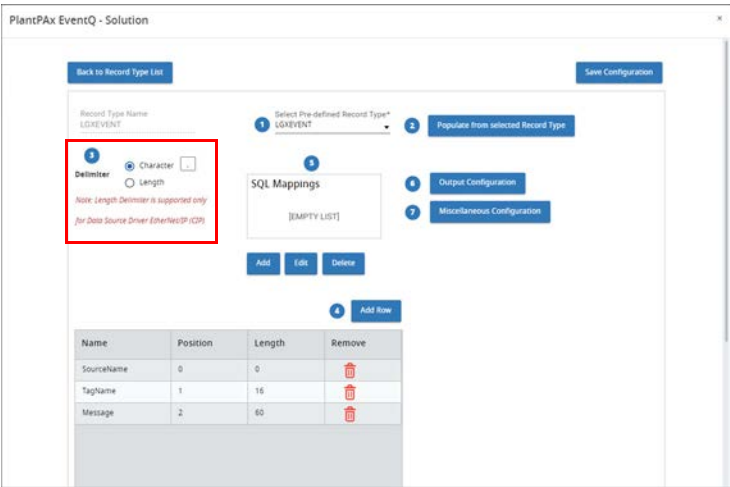
3. Use the “Select Pre-defined Record Types” drop-down and select a record type. (Example: LGXEVENT)
Supported Pre-Defined Record Types are:
 - LBSM (Restricted to Character Delimiter only)
 - RAC_EVENT
 - LGXEVENT

4. Select the “Populate from selected Template” button. This step loads the Record field in the Record Grid.

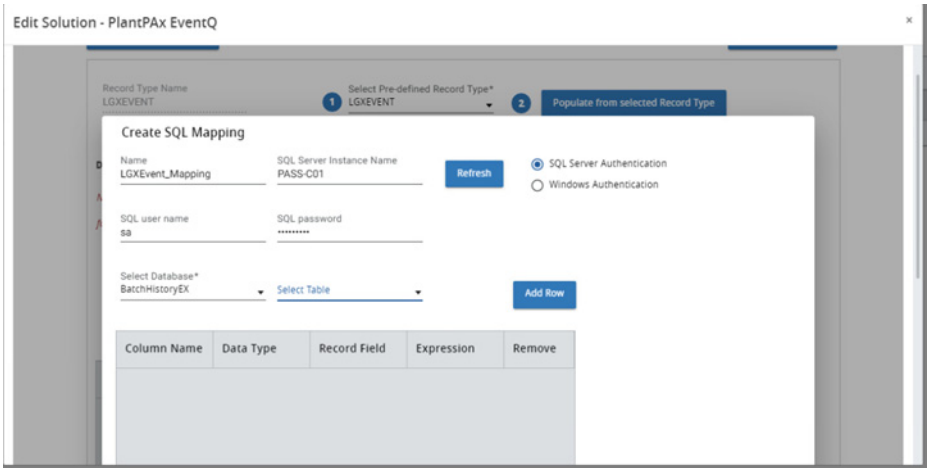
5. This step provides an option for the delimiter type, either “Character” or “Length”. (Example: character is a period)
User must configure/check Position column for Character delimiter type.
User must configure/check Length column for Length delimiter type.



For length delimiter, RAC_EVENT and LGXEVENT record types have predefined length values in the grid, for the other record type user must define its length values.



- 6. This step is optional. You can Add, Edit, and Delete the SQL Mapping for the processed Record.
- 7. Select the Add button for the SQL Mapping. A create SQL Mapping window launches.
- 8. Provide the Name of the Mapping and SQL Server details with the authentication method of choice.
- 9. Select the Refresh button to connect to the Database and load a list of available Databases into the Databases drop down. Selecting the Database loads the list of available SQL databases.



10. Select the "Create Table" option. This option is only for the Pre-Defined Record Type. Using this option, a Table is created in the selected Database with the name "<RecordTypeName>Data", and the data grid is auto populated with table columns and its mapping. Select update to finalize the changes and that redirects to the previous screen.

PlantPAx EventQ - Solution

Create SQL Mapping

Name: LGXEVENT_MAPPING SQL Server Instance Name: PPAIOEWS1 Refresh SQL Server Authentication Windows Authentication

SQL user name: sa SQL password: [password]

Select Database*: BatchHistoryEx Select Table: LGXEVENTData Add Row

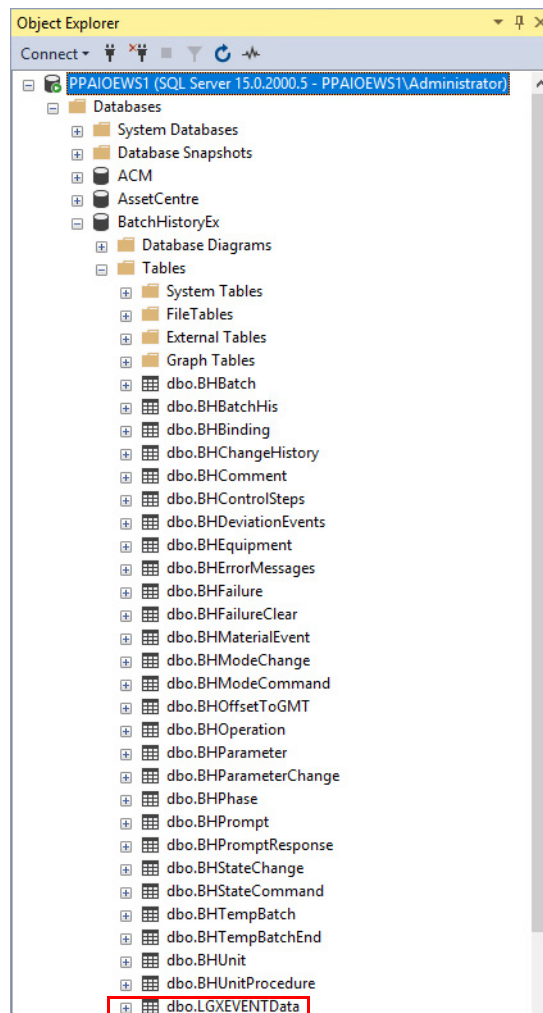
Column Name	Data Type	Record Field	Expression	Remove
TicksTimeStamp	String		TicksTimeStamp	
EventTimeStamp	String		EventTimeStamp	
RecType	String		RecordType	
SourceName	String	SourceName		
TagName	String	TagName		
Message	String	Message		

Populate from Table Create Table Cancel Update

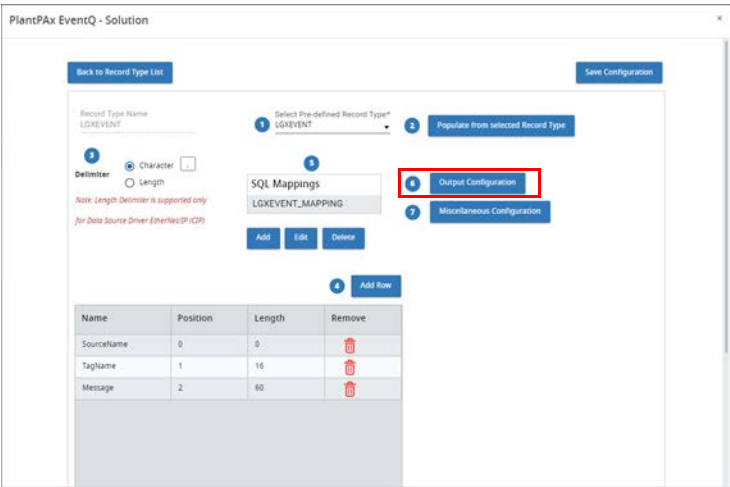


You can also select the table and use the option "Populate from table" to load the columns of the existing table. In this case, you map table columns to the Record Field and Expression. "Populate from table" is used in the case of User-Defined Record Type.

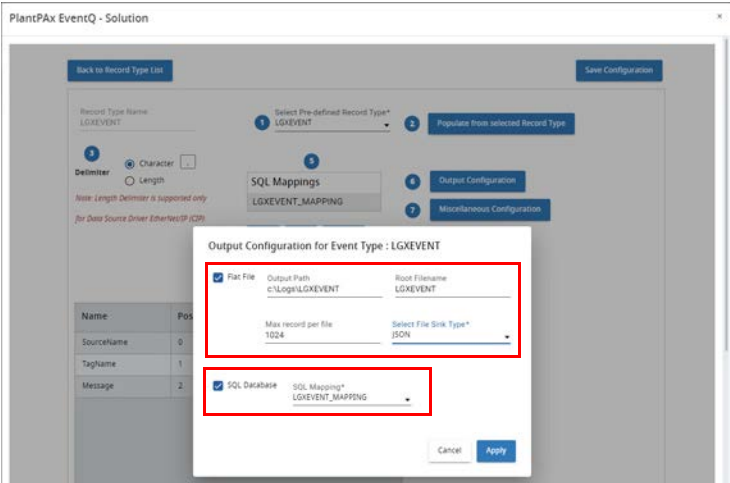
The following screenshot is from SQL Server Management Studio showing an example of the created table in the database.



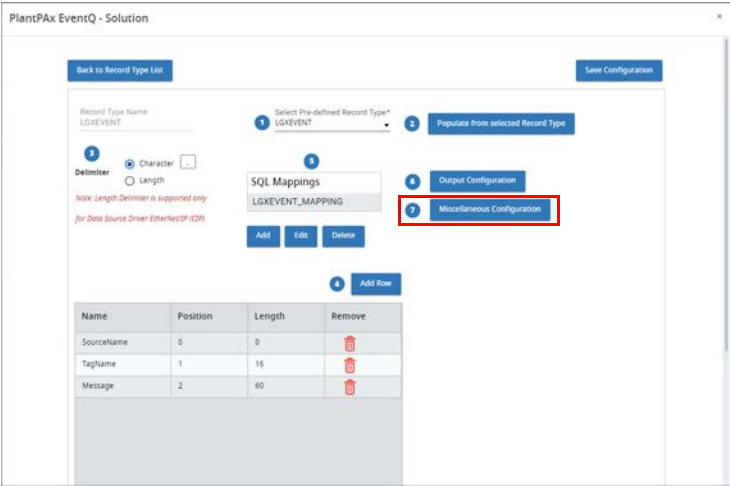
11. Select Output Configuration to configure the sink details.



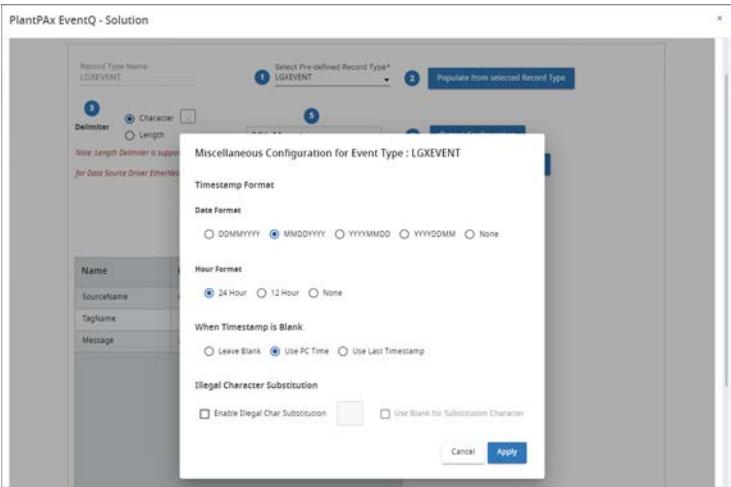
12. You can configure Flat File and/or SQL Database option as sink and select Apply.



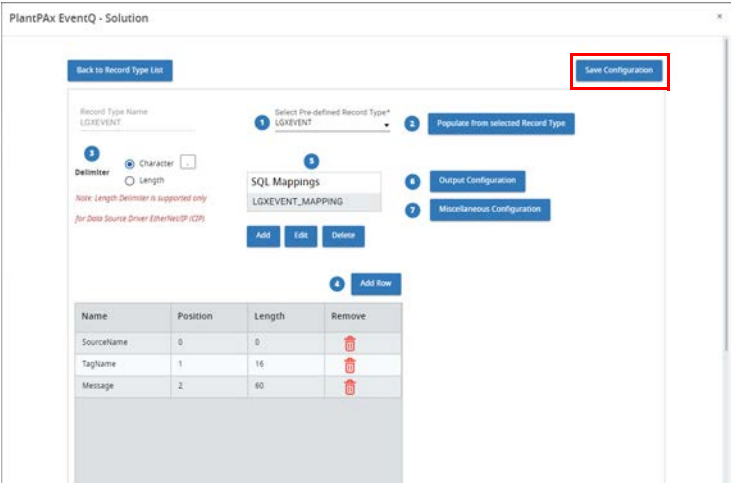
The final step for Miscellaneous Configuration is to configure the format of date and time and "Illegal Character Substitution" options.



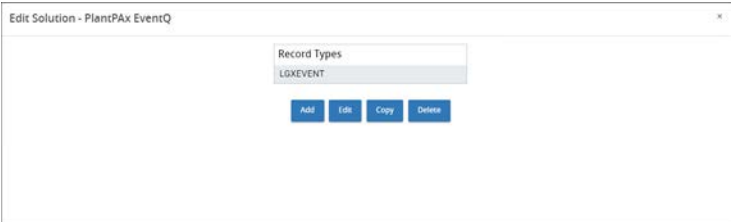
13. Select the appropriate format from the available options. The default value of Date format is “MMDDYYYY” and for Hour format “24” Hour.



14. Once the configuration is done, select “Save Configuration”, the configuration is saved to the Broker Add-On.



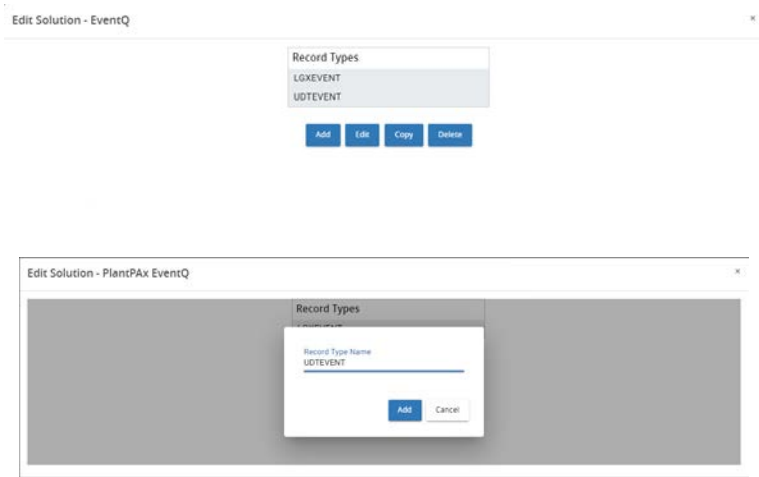
15. Start PlantPax EventQ Application, See [Start PlantPax EventQ Solution Application on page 110](#).
16. Once the Record Type configuration is saved, it is listed in the Record Types list.



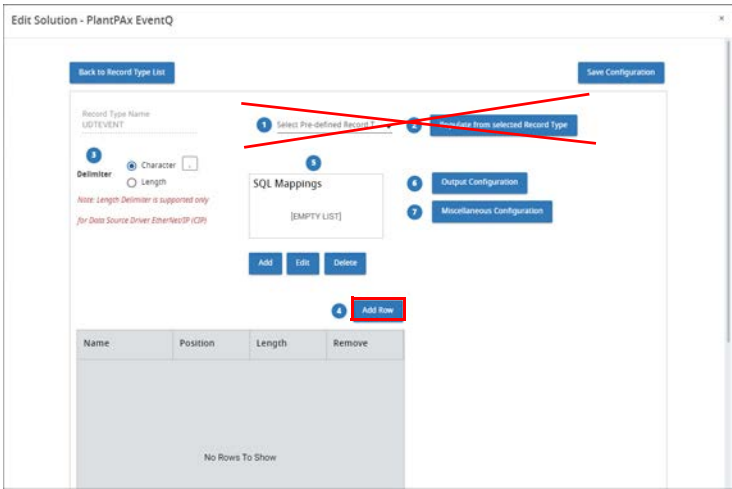
Add User-Defined Record Type (Optional)

User-defined record types are provided for users desiring different or additional table values. This document provides a user-defined record type overview, but this configuration is not used within the reporting example.

1. Select the Add button to add the new Record Type. A popup appears, provide the name of the Record Type, and select Add.

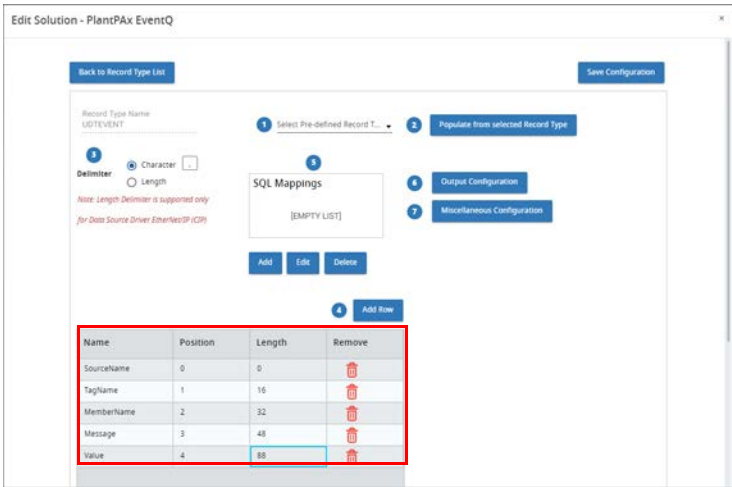


2. In the next view, you can add the rows using "Add Row".



IMPORTANT Select Pre-defined Record Types can be used for Pre-Defined Record Types only.

3. Repeat Add Row until all data fields are added.



The Broker-addon PlantPAx EventQ solution must know where data fields are located in the block of string data that is taken from the event queue. This can be determined in one of two ways:

- a. If the data is character delimited, the position is always used. The first token in the delimited string is position 1, the next position 2 and so-on. It is important to note that the position number starts at zero and not one. When character delimiting is selected, the delimiter may be configured. Usually, this character is a comma.

PPAX, XV101, PCmd_Pos2, Program Command to Open, 1

PPAX	XV101	PCmd_Pos2	Program Command to Open	1
Position 0	Position 1	Position 2	Position 3	Position 4

- b. If the data is length delimited, the length of the data field is used. The starting position of the data field is always the combined length of all fields before it. So, if the first field is three characters long, the first field starts at position zero in the string and spans three characters. The second field 12 character long. This field starts at position three in the string and spans 12 characters, making the third field, no matter how long it is, start at position 15 in the string.

Name	Data Type	Bytes	Length
SourceName	STRING_12	16 bytes	0
TagName	STRING_12	16 bytes	0 + 16 = 16
MemberName	STRING_12	16 bytes	16 + 16 = 32
Message	STRING_40	40 bytes	32 + 16 = 48
Value	STRING_8	12 bytes	48 + 40 = 88

4. You can Add, Edit, and Delete the SQL Mapping for the processed Record. Select the Add button and a popup is loaded for the SQL Mapping.

5

SQL Mappings

[EMPTY LIST]

Add
Edit
Delete

Edit Solution - PlantPAx EventQ

Back to Record Type List
Save Configuration

Record Type Name: UDTEVENT

1 Select Pre-defined Record Type
2 Populate from selected Record Type

Create SQL Mapping

Name

SQL user name

Select Database

SQL Server Instance Name

SQL password

Select Table

Refresh

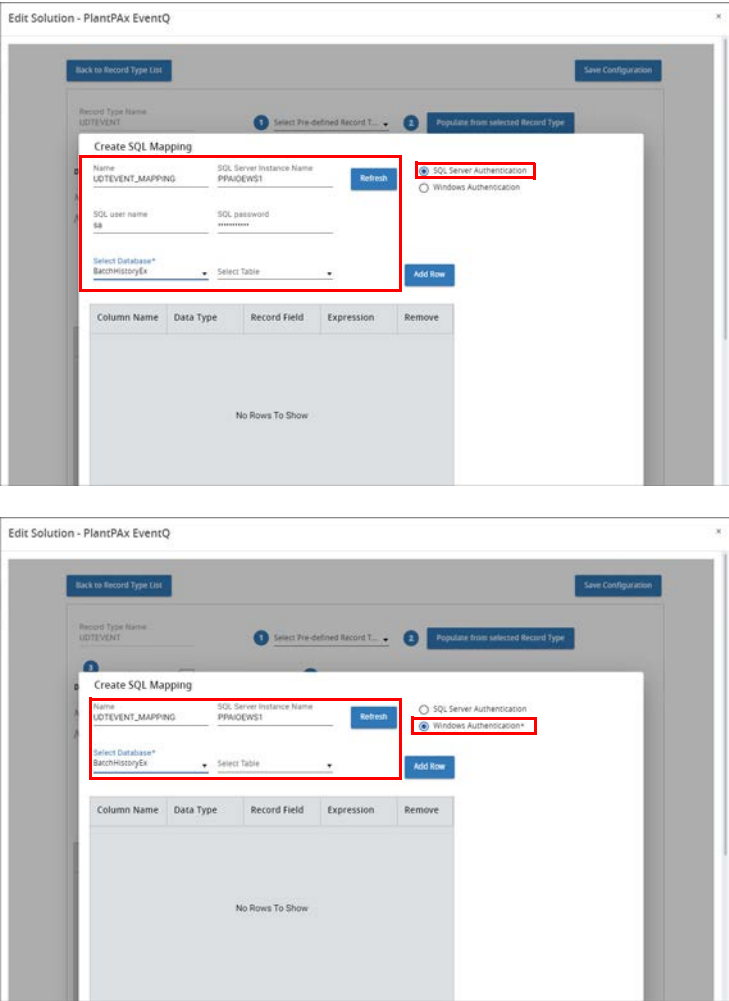
☒ SQL Server Authentication

☐ Windows Authentication

Add Row

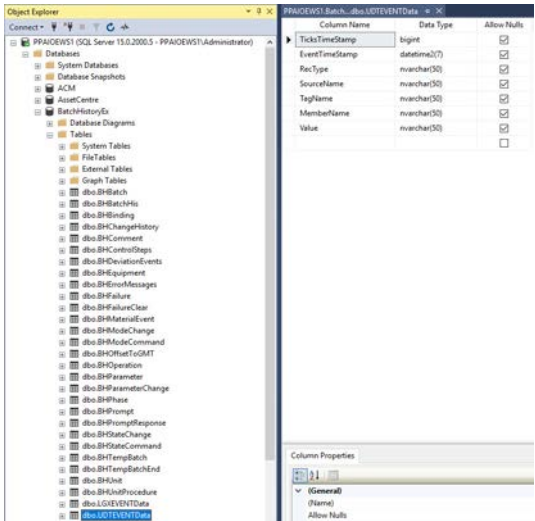
Column Name	Data Type	Record Field	Expression	Remove
No Rows To Show				

5. Provide the Name of the Mapping and SQL Server details, select the Refresh button to connect to the Database and load a list of available Databases into the Databases drop down. Selecting the Database loads the list of available tables.



IMPORTANT Add your users (Broker-addon running as a service requires NT AUTHORITY\Network Service for Windows Authentication) to database permissions and granting at least connect, execute, and Insert permissions. Windows Authentication will not work if SQL database is on another machine due to Broker-addon running as a service with Network Service account.

6. Select the table and use the option “Populate from table” to load the columns of an existing table. Map the table columns to the Data Type, Record Field, and Expression.



Create SQL Mapping - Populate from Table

Edit Solution - PlantPax EventQ

Create SQL Mapping

Name

UDTEVENT_MAPPING

SQL Server Instance Name

PPAIDEVST1

Refresh

☒ SQL Server Authentication

☐ Windows Authentication

SQL user name

sa

SQL password

Select Database*

BatchHistoryEx

Select Table*

UDTEVENTData

Add Row

Column Name	Data Type	Record Field	Expression	Remove
No Rows To Show				

Populate From Table

Create Table

Cancel

Update

Populate from Table for User-Defined Record Types - Data Type

Edit Solution - PlantPax EventQ

Create SQL Mapping

Name

UDTEVENT_MAPPING

SQL Server Instance Name

PPAIDEVST1

Refresh

☒ SQL Server Authentication

☐ Windows Authentication

SQL user name

sa

SQL password

Select Database*

BatchHistoryEx

Select Table*

UDTEVENTData

Add Row

Column Name	Data Type	Record Field	Expression	Remove
TicksTimeStamp	String			
EventTimeStamp	Numeric			
RecType				
SourceName				
TagName				
MemberName				
Message				
Value				

Populate From Table

Create Table

Cancel

Update

Populate from Table for User-Defined Record Types - Expression

Edit Solution - PlantPax EventQ

Create SQL Mapping

Name

UDTEVENT_MAPPING

SQL Server Instance Name

PPAIDEVST1

Refresh

☒ SQL Server Authentication

☐ Windows Authentication

SQL user name

sa

SQL password

Select Database*

BatchHistoryEx

Select Table*

UDTEVENTData

Add Row

Column Name	Data Type	Record Field	Expression	Remove
TicksTimeStamp				
EventTimeStamp				
RecType				
SourceName				
TagName				
MemberName				
Message				
Value				

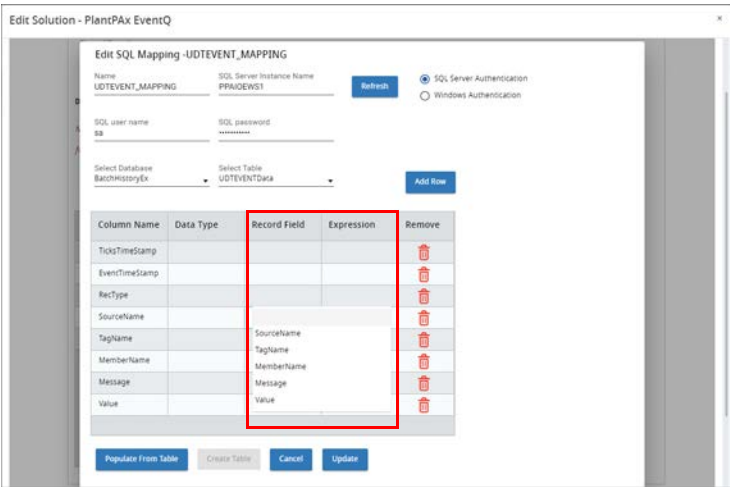
Populate From Table

Create Table

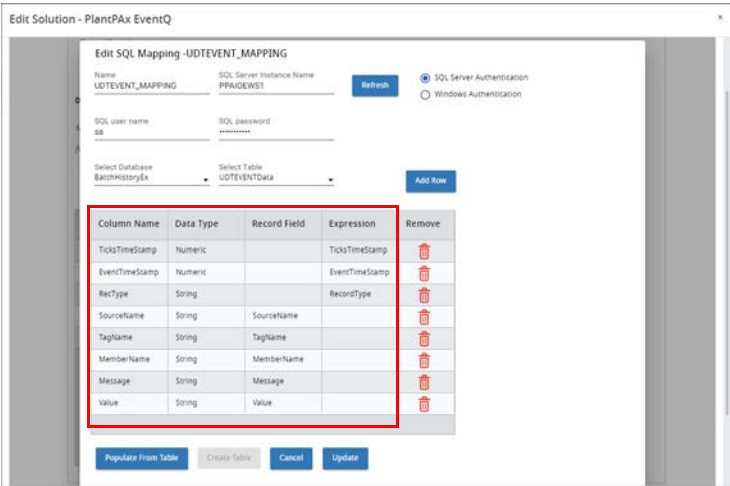
Cancel

Update

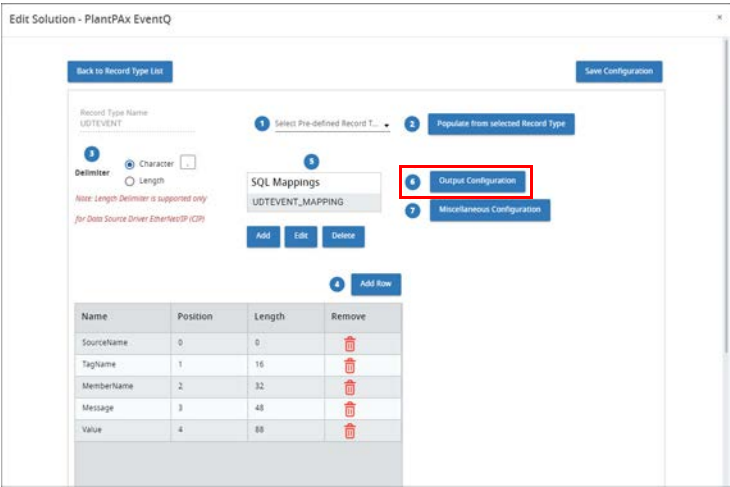
Populate from Table for User-Defined Record Types - Record Field



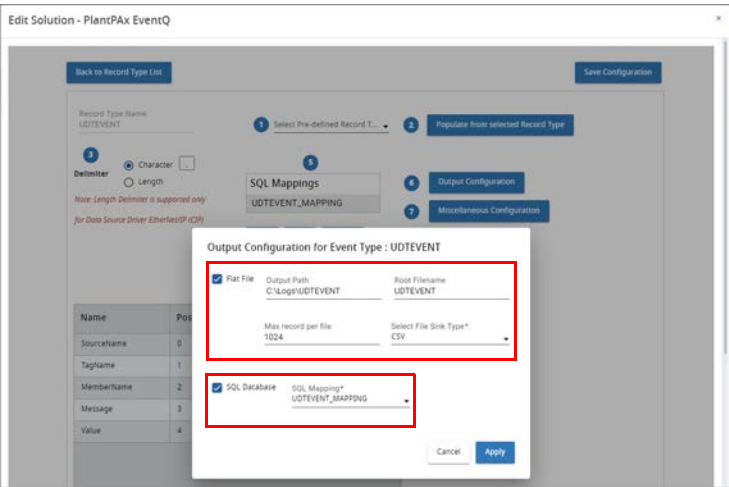
7. Once SQL mapping is done, select Update.



8. Click the Output Configuration to configure the sink details.



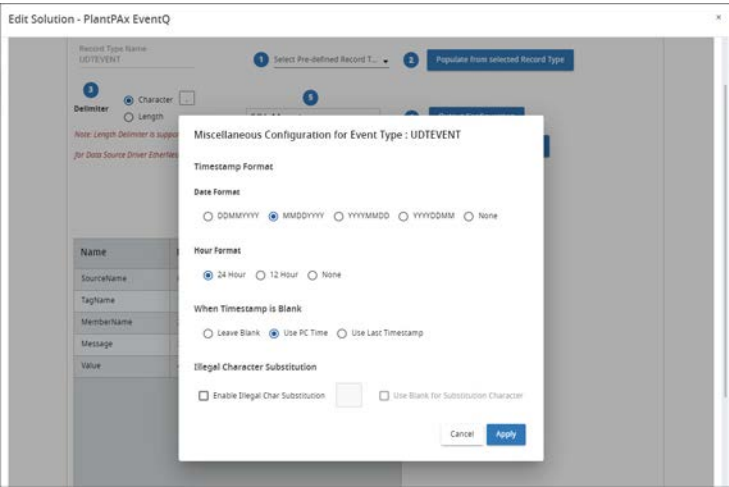
9. You can configure Flat File and/or SQL Database option as sink and select Apply.



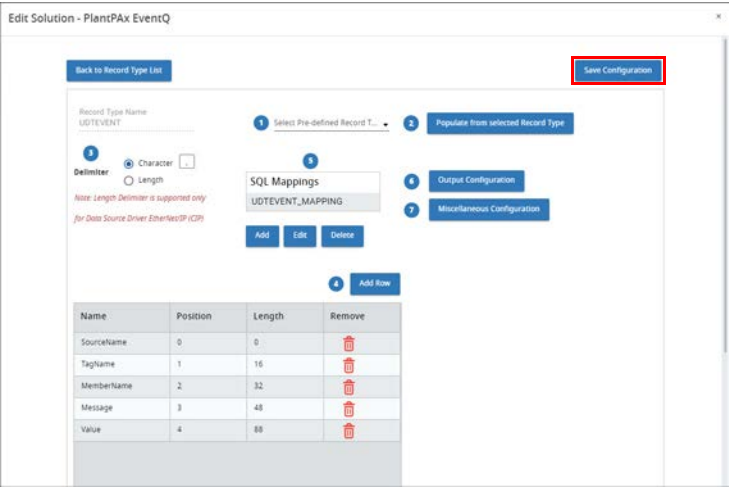
10. Select Miscellaneous Configuration to configure the format of date and time and “Illegal Character Substitution” options.

11. Select the appropriate format from the available options.

Default value of Date format is “MMDDYYYY” and for Hour format “24” Hour.

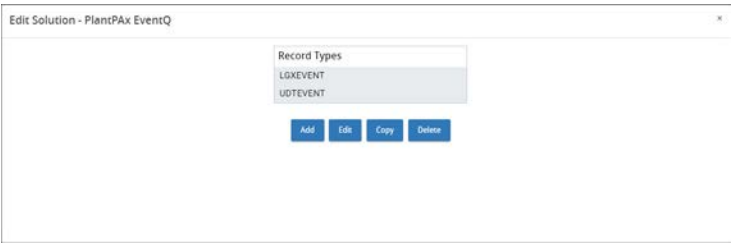


12. Once the configuration is done, select “Save Configuration” button, which saves the configuration to the Broker Add-On.



13. Start PlantPax EventQ Application, See [Start PlantPax EventQ Solution Application on page 110](#).

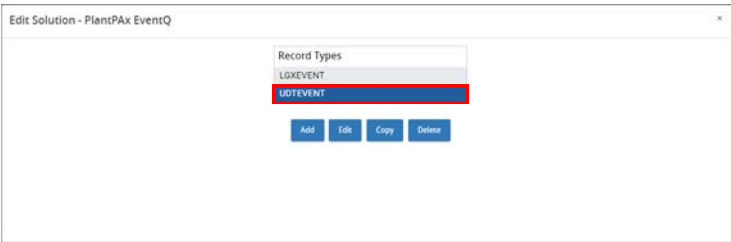
14. Once the Record Type configuration is saved, it is listed in the Record Types list.



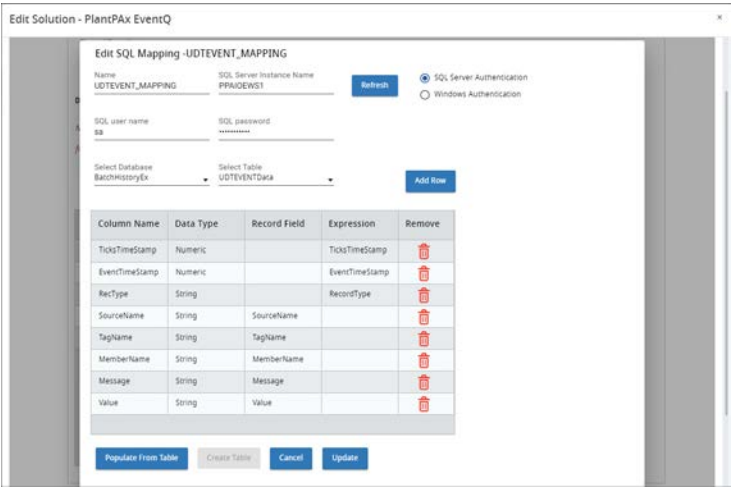
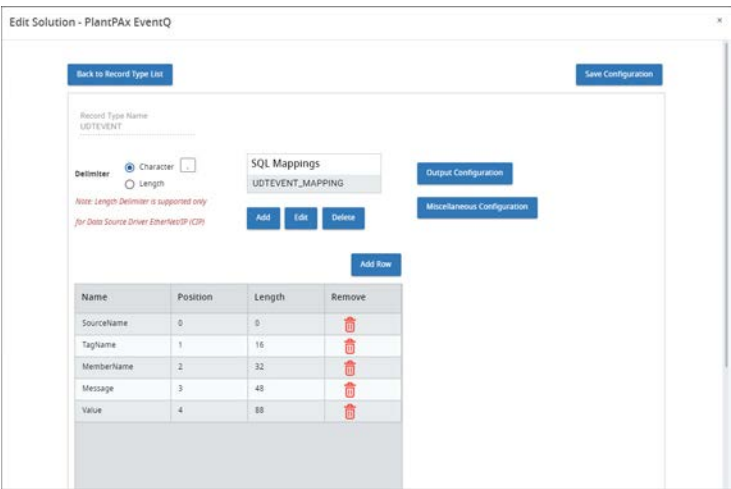
Edit Record Type

Use the following steps if you want to modify a record type.

- 1. Select the Record Type from available list of Record Types and select the Edit button.



- 2. You can edit/add any configuration such as Record Field Name, SQL Mapping, DB Table, Output Configuration, Miscellaneous Configuration etc.

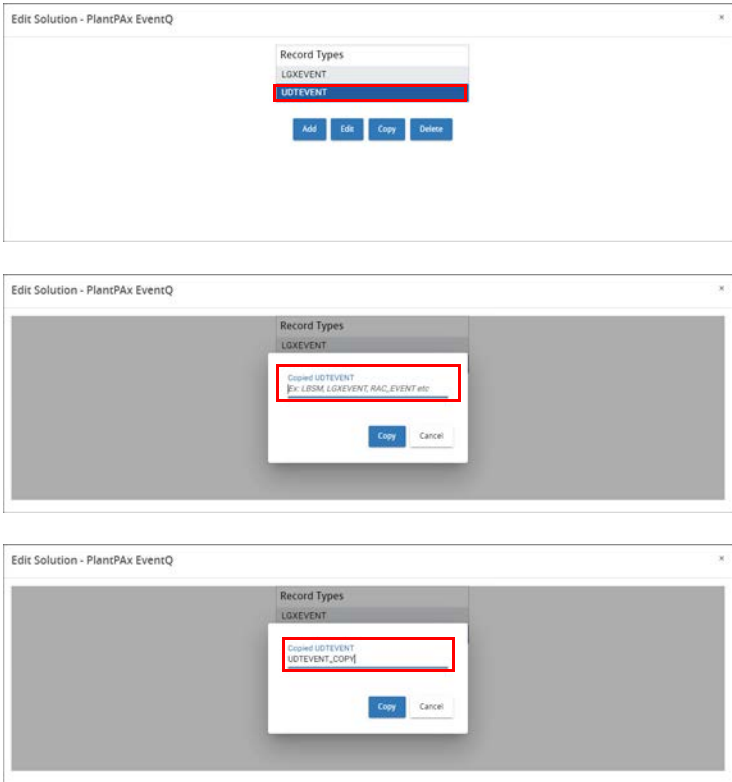


- 3. Selecting “Save Configuration”, will update the configuration and save to the Broker Add-On.

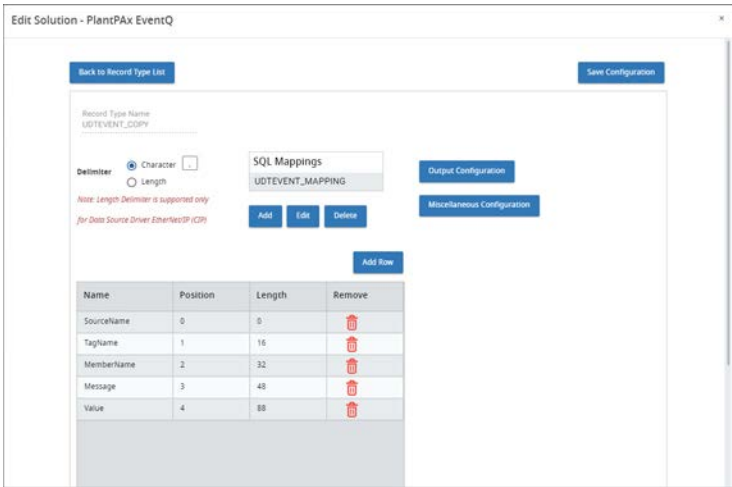
Copy Record Type

Use the following steps if you want to copy an existing record type to create a new record type.

- 1. To copy the Record Type, select the Record Type from available list of Record Types and select the “Copy” button.
Config UI creates a copy of the Configuration and attaches it to the new Record Type.



- 2. Control is redirected to the Record details screen with the pre-populated values copied from the other existing Record Type.
Verify the details and edit any configuration as per requirements.

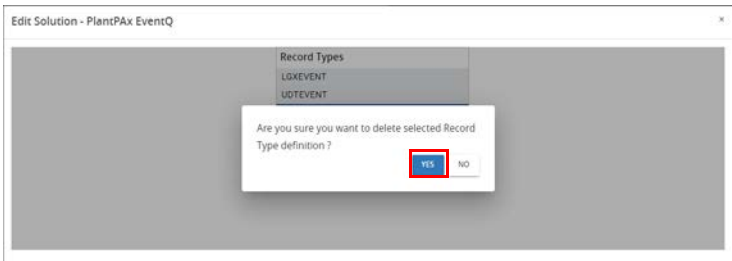
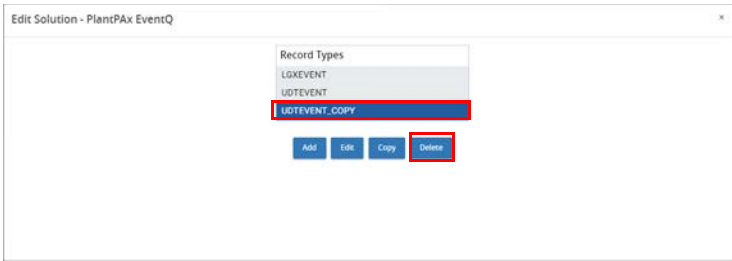


- 3. Once the configuration is verified/edited, select “Save Configuration” to update the configuration and save to the Broker Add-On.

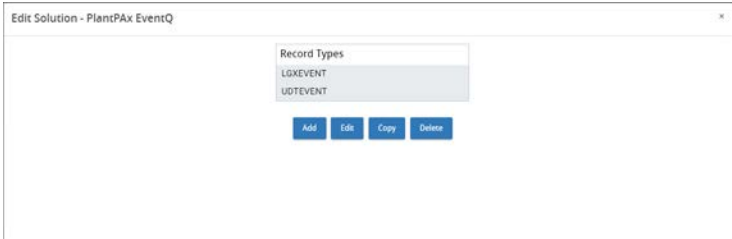
Delete Record Type

Use the following steps if you want to delete a record type.

- 1. Select the Record Type from the available list of Record Types and select the “Delete” button.



- 2. The Record Type and its configuration are deleted.



Event Data Processing and Persistence to Sink

Start PlantPAX EventQ Solution Application

Once a Record Type has been configured, you can Start the PlantPAX EventQ solution to process and store the Record Types into the specified sink type.

Select the play button (triangle icon) to start the PlantPAX EventQ solution.

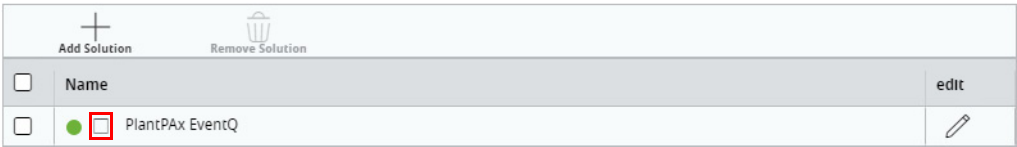
Configure Solutions

Add Solution		Remove Solution	
<input type="checkbox"/>	Name		edit
<input type="checkbox"/>	● ▶ PlantPAX EventQ		

Stop PlantPAx EventQ Solution Application

You can Stop the PlantPAx EventQ solution by selecting the stop button (square icon).

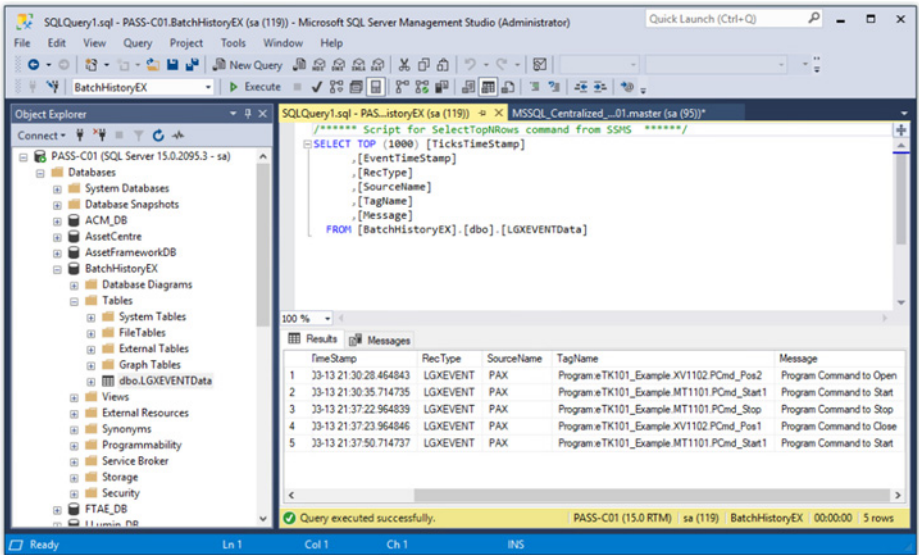
Configure Solutions



SQL Server EventQ Database

If your data sink is a SQL database, you can browse the table to confirm that your EventQ database is storing records.

1. Open SQL Server Management Studio.
2. In the Object Explorer, expand the Databases folder.
3. Locate your specified database and expand the tables folder within.
4. Right-click the table that is named LGXEVENTData and Select Top 1000 Rows.
5. The query window displays any recently recorded events.



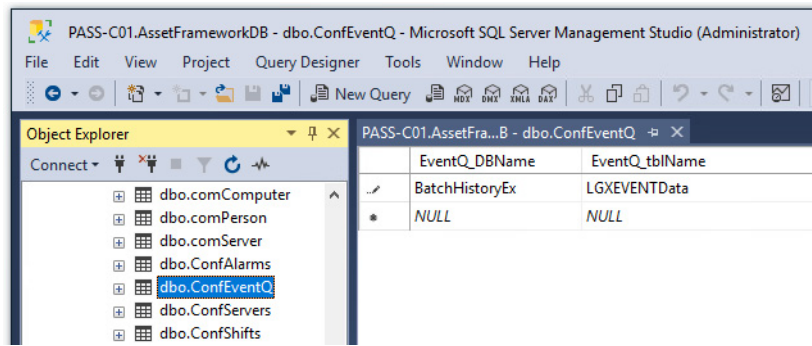
EventQ Reporting

Rockwell Automation Standard Reports and ThingWorx® Maintenance Reports can display EventQ data in the Object Event Reports if the EventQ solution writes to a SQL database on the same SQL server containing the centralized database.

Use the following steps to configure the centralized database to read from the EventQ solution database.

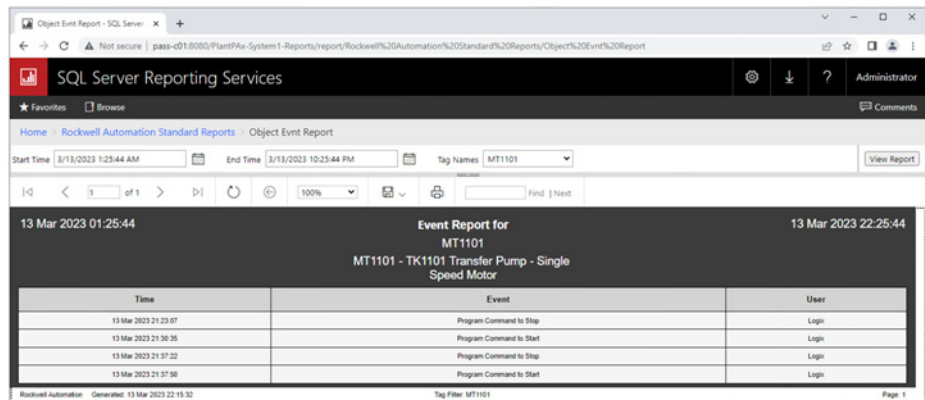
1. Launch the SQL Management Studio software and connect to your SQL server.
2. From the Object Explorer, expand the "Databases" folder, expand the centralized database folder (Example AssetFrameworkDB) and expand the tables folder.
3. Locate the table that is named dbo.ConfEventQ then right-click and select "Edit Top 200 Rows".

4. Type in your EventQ database name and EventQ database table name.

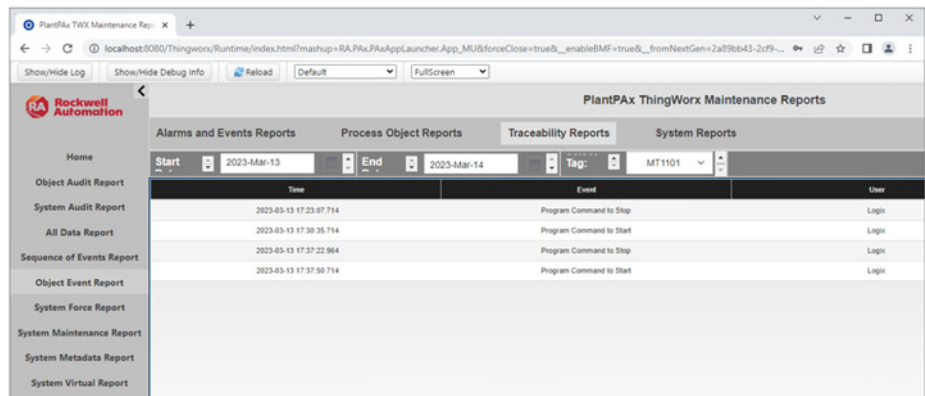


The SQL Agent Jobs are now able to process the EventQ data and display it on the Object Event Reports.

The following is an example of the Standard Reports Object Event Report. Notice the event for Program Command to Start was executed by the user "Logix", meaning the process controller code commanded the motor MT1101 to stop and start at four times.



The following is the same example using the ThingWorx Maintenance Repots > Traceability Reports > Object Event Report.



Record Types

The following examples show a record type displayed in various formats.

Event Data Samples persisted to various Sinks (Ex: RAC_EVENT)

Data Persisted to SQL Database

Results		Messages				
	TicksTimeStamp	EventTimeStamp	RecType	SourceName	TagName	Message
1	637854080764579968	2022-04-12 20:54:36.4582520	LGXEVENT	PPAX	MT104.PCmd_Start1	Program Command to Start
2	637854080882080000	2022-04-12 20:54:48.2083890	LGXEVENT	PPAX	MT104.PCmd_Stop	Program Command to Stop
3	637854080932080000	2022-04-12 20:54:53.2084540	LGXEVENT	PPAX	MT104.PCmd_Start2	Program Command to Start Reverse
4	637854080994579968	2022-04-12 20:54:59.4585100	LGXEVENT	PPAX	MT104.PCmd_Stop	Program Command to Stop
5	637854081127079936	2022-04-12 20:55:12.7086980	LGXEVENT	PPAX	XV101.PCmd_Pos2	Program Command to Open
6	637854081187079936	2022-04-12 20:55:18.7087570	LGXEVENT	PPAX	XV101.PCmd_Pos1	Program Command to Close
7	637854081344579968	2022-04-12 20:55:34.4589330	LGXEVENT	PPAX	XV102.PCmd_Pos2	Program Command to Open
8	637854081402089984	2022-04-12 20:55:40.2090180	LGXEVENT	PPAX	XV102.PCmd_Stop	Program Command to Stop
9	637854081449590016	2022-04-12 20:55:44.9590510	LGXEVENT	PPAX	XV102.PCmd_Pos1	Program Command to Close
10	637854081499590016	2022-04-12 20:55:49.9591240	LGXEVENT	PPAX	XV102.PCmd_Stop	Program Command to Stop
11	637854081624589952	2022-04-12 20:56:02.4592560	LGXEVENT	PPAX	XY103.PCmd_On	Program Command On
12	637854081672089984	2022-04-12 20:56:07.2093330	LGXEVENT	PPAX	XY103.PCmd_Off	Program Command Off

Data Persisted to Flat File Format - CSV

```

1  TicksTimeStamp,EventTimeStamp,RecType,SourceName,TagName,Message
2  637854088124670000,2022-04-12 21:06:52.467014,LGXEVENT,PPAX,MT104.PCmd_Start1,Program Command to Start
3  637854088257170000,2022-04-12 21:07:05.717183,LGXEVENT,PPAX,MT104.PCmd_Stop,Program Command to Stop
4  637854088379670000,2022-04-12 21:07:17.967336,LGXEVENT,PPAX,MT104.PCmd_Start2,Program Command to Start Reverse
5  637854088504670000,2022-04-12 21:07:30.467460,LGXEVENT,PPAX,MT104.PCmd_Stop,Program Command to Stop
6  637854088627170000,2022-04-12 21:07:42.717623,LGXEVENT,PPAX,XV101.PCmd_Pos2,Program Command to Open
7  637854088707170000,2022-04-12 21:07:50.717740,LGXEVENT,PPAX,XV101.PCmd_Pos1,Program Command to Close
8  637854088844670000,2022-04-12 21:08:04.467865,LGXEVENT,PPAX,XV102.PCmd_Pos2,Program Command to Open
9  637854088902170000,2022-04-12 21:08:10.217972,LGXEVENT,PPAX,XV102.PCmd_Stop,Program Command to Stop
10 637854088972180000,2022-04-12 21:08:17.218054,LGXEVENT,PPAX,XV102.PCmd_Pos1,Program Command to Close
11 637854089024680000,2022-04-12 21:08:22.468090,LGXEVENT,PPAX,XV102.PCmd_Stop,Program Command to Stop
12 637854089287180000,2022-04-12 21:08:48.718414,LGXEVENT,PPAX,XY103.PCmd_On,Program Command On
13 637854089327180000,2022-04-12 21:08:52.718465,LGXEVENT,PPAX,XY103.PCmd_Off,Program Command Off

```

Data Persisted to Flat File Format - JSON

```

1  [{"TicksTimeStamp": 637854080764580000,
2    "EventTimeStamp": "2022-04-12 20:54:36.458252",
3    "RecType": "LGXEVENT",
4    "SourceName": "PPAX",
5    "TagName": "MT104.PCmd_Start1",
6    "Message": "Program Command to Start"
7  }, {
8    "TicksTimeStamp": 637854080882080000,
9    "EventTimeStamp": "2022-04-12 20:54:48.208389",
10   "RecType": "LGXEVENT",
11   "SourceName": "PPAX",
12   "TagName": "MT104.PCmd_Stop",
13   "Message": "Program Command to Stop"
14 }, {
15   "TicksTimeStamp": 637854080932080000,
16   "EventTimeStamp": "2022-04-12 20:54:53.208454",
17   "RecType": "LGXEVENT",
18   "SourceName": "PPAX",
19   "TagName": "MT104.PCmd_Start2",
20   "Message": "Program Command to Start Reverse"
21 }, {
22   "TicksTimeStamp": 637854080994580000,
23   "EventTimeStamp": "2022-04-12 20:54:59.458510",
24   "RecType": "LGXEVENT",
25   "SourceName": "PPAX",
26   "TagName": "MT104.PCmd_Stop",
27   "Message": "Program Command to Stop"
28 }, {
29   "TicksTimeStamp": 637854081127080000,
30   "EventTimeStamp": "2022-04-12 20:55:12.708698",
31   "RecType": "LGXEVENT",
32   "SourceName": "PPAX",
33   "TagName": "XV101.PCmd_Pos2",
34   "Message": "Program Command to Open"
35 }, {
36   "TicksTimeStamp": 637854081187080000,
37   "EventTimeStamp": "2022-04-12 20:55:18.708757",
38   "RecType": "LGXEVENT",
39   "SourceName": "PPAX",
40   "TagName": "XV101.PCmd_Pos1",
41   "Message": "Program Command to Close"
42 }, {
43   "TicksTimeStamp": 637854081344580000,
44   "EventTimeStamp": "2022-04-12 20:55:34.458933",
45   "RecType": "LGXEVENT",
46   "SourceName": "PPAX",
47   "TagName": "XV102.PCmd_Pos2",
48   "Message": "Program Command to Open"
49 }, {
50   "TicksTimeStamp": 637854081402090000,
51   "EventTimeStamp": "2022-04-12 20:55:40.209018",
52   "RecType": "LGXEVENT",
53   "SourceName": "PPAX",
54   "TagName": "XV102.PCmd_Stop",
55   "Message": "Program Command to Stop"
56 }, {
57   "TicksTimeStamp": 637854081449590000,
58   "EventTimeStamp": "2022-04-12 20:55:44.959051",
59   "RecType": "LGXEVENT",
60   "SourceName": "PPAX",
61   "TagName": "XV102.PCmd_Pos2",
62   "Message": "Program Command to Open"
63 }]
```


Data Persisted to Flat File Format - XML

```

1  <?xml version="1.0" encoding="utf-8"?>
2  <root>
3    <elements>
4      <element>
5        <TicksTimeStamp>637854091112200000</TicksTimeStamp>
6        <EventTimeStamp>"2022-04-12 21:11:51.220589"</EventTimeStamp>
7        <RecType>"LGXEVENT"</RecType>
8        <SourceName>"PPAX"</SourceName>
9        <TagName>"MT104.PCmd_Start1"</TagName>
10       <Message>"Program Command to Start"</Message>
11     </element>
12     <element>
13       <TicksTimeStamp>637854091152200000</TicksTimeStamp>
14       <EventTimeStamp>"2022-04-12 21:11:55.220638"</EventTimeStamp>
15       <RecType>"LGXEVENT"</RecType>
16       <SourceName>"PPAX"</SourceName>
17       <TagName>"MT104.PCmd_Stop"</TagName>
18       <Message>"Program Command to Stop"</Message>
19     </element>
20     <element>
21       <TicksTimeStamp>637854091194700000</TicksTimeStamp>
22       <EventTimeStamp>"2022-04-12 21:11:59.470665"</EventTimeStamp>
23       <RecType>"LGXEVENT"</RecType>
24       <SourceName>"PPAX"</SourceName>
25       <TagName>"MT104.PCmd_Start2"</TagName>
26       <Message>"Program Command to Start Reverse"</Message>
27     </element>
28     <element>
29       <TicksTimeStamp>637854091234700000</TicksTimeStamp>
30       <EventTimeStamp>"2022-04-12 21:12:03.470711"</EventTimeStamp>
31       <RecType>"LGXEVENT"</RecType>
32       <SourceName>"PPAX"</SourceName>
33       <TagName>"MT104.PCmd_Stop"</TagName>
34       <Message>"Program Command to Stop"</Message>
35     </element>
36     <element>
37       <TicksTimeStamp>637854091349700000</TicksTimeStamp>
38       <EventTimeStamp>"2022-04-12 21:12:14.970861"</EventTimeStamp>
39       <RecType>"LGXEVENT"</RecType>
40       <SourceName>"PPAX"</SourceName>
41       <TagName>"XV101.PCmd_Pos2"</TagName>
42       <Message>"Program Command to Open"</Message>
43     </element>
44     <element>
45       <TicksTimeStamp>637854091389700000</TicksTimeStamp>
46       <EventTimeStamp>"2022-04-12 21:12:18.970904"</EventTimeStamp>
47       <RecType>"LGXEVENT"</RecType>
48       <SourceName>"PPAX"</SourceName>
49       <TagName>"XV101.PCmd_Pos1"</TagName>
50       <Message>"Program Command to Close"</Message>
51     </element>
52     <element>
53       <TicksTimeStamp>637854091532210000</TicksTimeStamp>
54       <EventTimeStamp>"2022-04-12 21:12:33.221092"</EventTimeStamp>
55       <RecType>"LGXEVENT"</RecType>
56       <SourceName>"PPAX"</SourceName>

```

Data Persisted to Flat File Format - Raw Data

```

1  ("TicksTimeStamp":637854097299770000,"EventTimeStamp":"2022-04-12 21:12:09.977943","RecType":"LGXEVENT","SourceName":"PPAX","TagName":"MT104.PCmd_Start1","Message":"Program Command to Start")
2  ("TicksTimeStamp":637854097417280000,"EventTimeStamp":"2022-04-12 21:12:21.728097","RecType":"LGXEVENT","SourceName":"PPAX","TagName":"MT104.PCmd_Stop","Message":"Program Command to Stop")
3  ("TicksTimeStamp":637854097529780000,"EventTimeStamp":"2022-04-12 21:12:32.978214","RecType":"LGXEVENT","SourceName":"PPAX","TagName":"MT104.PCmd_Start2","Message":"Program Command to Start Reverse")
4  ("TicksTimeStamp":637854097647280000,"EventTimeStamp":"2022-04-12 21:12:44.728375","RecType":"LGXEVENT","SourceName":"PPAX","TagName":"MT104.PCmd_Stop","Message":"Program Command to Stop")
5  ("TicksTimeStamp":637854097772280000,"EventTimeStamp":"2022-04-12 21:12:57.722854","RecType":"LGXEVENT","SourceName":"PPAX","TagName":"XV101.PCmd_Pos2","Message":"Program Command to Open")
6  ("TicksTimeStamp":637854097832280000,"EventTimeStamp":"2022-04-12 21:12:03.228359","RecType":"LGXEVENT","SourceName":"PPAX","TagName":"XV101.PCmd_Pos1","Message":"Program Command to Close")
7  ("TicksTimeStamp":637854097947280000,"EventTimeStamp":"2022-04-12 21:12:14.728730","RecType":"LGXEVENT","SourceName":"PPAX","TagName":"XV101.PCmd_Pos2","Message":"Program Command to Open")
8  ("TicksTimeStamp":637854097997280000,"EventTimeStamp":"2022-04-12 21:12:19.728805","RecType":"LGXEVENT","SourceName":"PPAX","TagName":"XV102.PCmd_Pos2","Message":"Program Command to Stop")
9  ("TicksTimeStamp":637854098039780000,"EventTimeStamp":"2022-04-12 21:12:23.978822","RecType":"LGXEVENT","SourceName":"PPAX","TagName":"XV102.PCmd_Pos1","Message":"Program Command to Close")
10 ("TicksTimeStamp":637854098087280000,"EventTimeStamp":"2022-04-12 21:12:28.728858","RecType":"LGXEVENT","SourceName":"PPAX","TagName":"XV102.PCmd_Pos2","Message":"Program Command to Stop")
11 ("TicksTimeStamp":637854098124790000,"EventTimeStamp":"2022-04-12 21:12:31.479055","RecType":"LGXEVENT","SourceName":"PPAX","TagName":"XV103.PCmd_On","Message":"Program Command On")
12 ("TicksTimeStamp":637854098249790000,"EventTimeStamp":"2022-04-12 21:12:34.497903","RecType":"LGXEVENT","SourceName":"PPAX","TagName":"XV103.PCmd_Off","Message":"Program Command Off")

```

Notes:

Troubleshooting

About

Use this section to troubleshoot issues that you might run into while deploying these reports. This section of the document should be used if issues arise during the installation or connection steps.

For issues that might not be included in this section, please visit the Rockwell Automation Knowledgebase to look for additional solutions. If no online solution exists, please contact Rockwell Automation Technical Support for further assistance.

SQL / Historian Issues

Issue	Suggestion
Using the Microsoft® SQL Server Studio Manager to test the linked server connection to the Historian fails with error "Cannot initialize the data source object of OLE DB provider "PIOLEDB" for linked server "PI_Link". OLE DB provider for linked server "PI_Link" returned message "[PI SDK] Unable to open a session server". [-10723] PINET: "No Connection". (Microsoft SQL Server, Error: 7303)	<ul style="list-style-type: none"> • Confirm the required SQL and Historian TCP ports are allowed in the Windows® firewall. • Ensure the time zone, current time, and date of all server computers match. • Ensure that the Historian server has an OLEDB provider license. • On the Historian computer, open Windows "Services" and restart the "PI OLEDB Enterprise Agent" service. • On the Historian computer, open Windows "Services" and configure the "PI Licensing" service as automatic/delayed then restart it.
Using the Microsoft SQL Server Studio Manager to test the linked server connection to the Historian fails with error "Cannot create an instance of OLE DB provider "PIOLEDBENT" for linked server "PI_Link". (Microsoft SQL Server, Error: 7302)	<p>The Allow InProcess option must be set on the specific provider before the linked server is created. You can add the linked server provider option in SQL Server Management Studio using this procedure:</p> <p>Navigate to [SQL Server Database]--> Server Objects--> Linked Servers-> Providers. Right-click on the provider (PI-OLEDB), and select Properties. Check the Allow InProcess option to enable the property.</p>
Using the Microsoft SQL Server Studio Manager to test the linked server connection to the Historian fails with error "Cannot initialize the data source object of OLE DB provider "PIOLEDB" for linked server "PI_Link". OLE DB provider for linked server "PI_Link" returned message "[PI SDK] Registry item not found in collection". (Microsoft SQL Server, Error: 7303)	<p>On the Historian computer, open Windows "Services" and restart the "PI OLEDB Enterprise Agent" service.</p>
Using the Microsoft SQL Server Studio Manager to test the linked server connection to the Historian fails with error "Cannot initialize the data source object of OLE DB provider "PIOLEDB" for linked server "PI_Link". OLE DB provider for linked server "PI_Link" returned message "[PI SDK] The requested server was not found in the known server table. HIST-01". (Microsoft SQL Server, Error: 7303)	<p>Seen on domain computers where the linked server data source contained a computer name that is not specified with the fully qualified domain name (FQDN).</p> <p>Rerun the StandardReports.BAT file and enter the Historian server name using the fully qualified domain names (FQDN) or manually recreate the linked server with the following parameters:</p> <p>Linked Server: PI_Link Provider: PI OLE DB Provider Data Source: HIST-01.PlantPax.Local String Provider: Integrated Security=SSPI Catalog: piarchive Security tab: Be made using the login's current security context. Server Options tab: Collation Compatible True, Connection Timeout 30, Query Timeout 600</p>

Standard Report Issues

Issue	Suggestion
How to update an existing standard report installation to a new version?	<p>Since the source databases are only read from, the current procedure is to delete the centralized database, SQL Agent Jobs, delete the three datasource on SSRS reports and delete the PI Views.</p> <p>Then run the new StandardReports.BAT file while following the steps in the install standard reports section of this document.</p>
How are min/max/average values calculated and used?	The Asset Framework calculates min/max/average for the last hour on a schedule every hour. These values are then stored back into Historian as their own separate points. SQL Agent jobs run stored procedures within the centralized database that read the Asset Framework values every hour and saves that for each hourly interval. SQL stored procedures also use these hourly values to calculate min/max/average for shift and daily values.
How do I configure my shift times (morning/afternoon/evening) to my local time?	Shift times are set to UTC time in a table within the centralized database. Modification of time zones is prevented due to a current anomaly in process library version 5.10 and prior..
.NET Framework Error during the "Publish Reports" step.	<p>If an additional window appears prompting for .NET Framework, ensure that the required version is installed.</p> <ul style="list-style-type: none"> • SQL Server 2014 and SQL Server 2012 use .NET Framework 3.5 SP1 • SQL Server 2016 (13.x) and later require .NET Framework 4.6 or greater
StandardReport_5.xx.xx.BAT indicates an issue with 'rs.exe' is not recognized as an internal or external command, operable program, or batch file.	<p>Ensure the StandardReports.BAT file was modified per SQL Server version, as described in the installing standard reports section of this document.</p> <p>Verify that standard ASCII quotes are around the path to RS.exe.</p>
StandardReport_5.xx.xx.BAT indicates an issue after attempting to deploy reports "Could not connect to server: http://localhost:8080/ReportServer/ReportService2005.asmx	<p>Ensure that the complete process library directory structure was copied or accessible.</p> <p>Configure a temporary execution account within SSRS to match the credentials running the batch file script.</p>

ThingWorx Foundation Issues

Issue	Suggestion
ThingWorx® Foundation installer error: There has been an error. Java keytool binary is not found in \$PATH. Please add it the \$PATH	Verify %java_home%\bin has been added to the Path for System Environment variables
OPC Aggregator Project needs refreshed and you can't restart the server.	Select Runtime > Reinitialize from the top menu bar. This option reinitializes the runtime service and the event log displays new connection events for ThingWorx and OPC UA Client.
OPC Aggregator Project configuration ThingWorx Native Interface. Connection to ThingWorx failed. Platform = localhost:8080/Thingworx/WS, error = could not initialize a secure socket connection.	Use the IP address of the ThingWorx server for the project host setting.
OPC Aggregator Device Import Items Failed to connect to UA Server 'opc.tcp://hostname:portnumber/endpointname' for browsing	Launch FactoryTalk® Linx Gateway Configuration and verify the following: <ul style="list-style-type: none"> • UA Server and Discovery services are running • Windows Defender Firewall allows TCP traffic for the designated port • Security option that is selected for the UA Server Endpoints matches for both the Server and Client sides. (Example: both set to None, Basic256, and so on)
Control Strategy Health Dashboards are not updating with live data	<ol style="list-style-type: none"> 1. Ensure the process object tags have been generated in the OPC Aggregator before the dashboard things CSV file was imported. If not generated before import, then generated in the OPC Aggregator, delete the dashboard thing in ThingWorx Composer™ and reimport the dashboard. If the tags were generated before import, then use the OPC Aggregator software as a test client to see live tag values. Expand you device to the tag level and right-click on the tag of interest. Select the option "Launch Quick Client Here". An OPC UA test client will launch and allow you to browse live tags, before they enter ThingWorx. Fix any OPC connection errors here before focusing on ThingWorx. 2. Browse "Things" with ThingWorx Composer. Find the dashboard thing that isn't updating. (Example "LT1101_Remote_Thing". Select the thing and select the Properties and Alerts section. Hover your mouse over one of the Sources. A tooltip window will appear and display the full path back to the controller. Examine if the path is correct. (Example: FTLGW.PAX.OPC-UA_Server_PlantPax_Data_Area.PAX.Online.Program:eTK101.Example.LT1101.SrcQ) If the path is missing the LT1101 name and only lists .SrcQ then it's likely the thing was created before the OPC Aggregator had tags generated for ThingWorx. Scroll towards the bottom of the Properties and Alerts section. Verify that the "isConnected" property has a checkbox for the value. If not, ThingWorx is not connecting to the OPC Aggregator.

Vuforia Studio Issues

Issue	Suggestion
Online Installer - Studio Setup. An error occurred attempting to install Studio. Log file indicates URLDownloadToCacheFile failed with HRESULT '-2146697208' An error occurred try to download 'https://studio-download.vuforia.io/downloads/Studio application'.	Copy the URL into a web browser to download the installer. https://studio-download.vuforia.io/downloads/Studio application
Vuforia® Experience Service start-es.bat batch file disappears during loading	Edit the C:\ptc\studio-es\bin\start-es.bat file by putting "cmd /k" at the end of the file to keep the window open after error. Then run the start-es.bat file again and look for authentication fatal errors. Client authentication is controlled by a configuration file, which traditionally is named pg_hba.conf and is stored in the database cluster's data directory. Verify PostgreSQL pg_hba.conf file is configured for md5 (Secure) or trust (not Secure) for the connection. Restart the PostgreSQL service and try running the start-es.bat file again. For Windows workgroup computers, use the fqdn name of .local in the hosts file.
Localhost:3000 says Error connecting to the ThingWorx server.	Verify that Windows firewall has open TCP ports for ThingWorx. Verify that Vuforia Experience Services is installed and running.
Localhost:3000 says Undefined	Click OK and verify that the External Data bindings exist as RemoteThings in ThingWorx Composer. This message will likely be seen after opening an Example Project provided with the Process Library.

EventQ Solution Issues

Issue	Suggestion
Buttons to add Data Source, Model, and Applications appear dimmed.	Change the FactoryTalk® Edge Gateway™ to Online mode.
FactoryTalk Edge Gateway - Unable to Go Online. A different configuration is running on the gateway.	Either upload the configuration from the gateway or download your current configuration to the gateway.
FactoryTalk Edge Gateway - Data Source Collection Path.	<ul style="list-style-type: none"> • The driver is offline, and the collection path is inactive. • Verify controller communications (EtherNet/IP™ or FactoryTalk Live Data) are active and working. • Verify that the controller contains a downloaded project.
FactoryTalk Edge Gateway application type of Add-On is missing.	Verify that the Broker-addon software is installed and confirm that Broker-Addon service is running.
The PlantPAx® Event solution is not connecting to your SQL database or can't create the database table.	Ensure that you add your users (Broker-addon running as a service requires NT AUTHORITY\Network Service for Windows Authentication) to database permissions and granting at least connect, execute, and Insert permissions. Windows Authentication will not work if SQL database is on another machine due to Broker-addon running as a service with Network Service account.
The Broker-addon doesn't have an Edit option.	Ensure that the Broker-addon Windows Service is running. Restart the service if the Broker-addon doesn't show a green running indicator.

Additional Resources

These documents contain additional information concerning related products from Rockwell Automation.

Resource	Description
PlantPax DCS - Configuration and Implementation Manual, Publication PROCES-UM100	Provides system guidelines and instructions to assist with the development of your PlantPax system. <ul style="list-style-type: none"> • Process Automation System Server chapter <ul style="list-style-type: none"> - Add the Alarms and Events Database section - Add a Data Server section - Add an Alarm Server section Ensure you enable historization on both the Data Server and the Alarm Server <ul style="list-style-type: none"> • Historical Data chapter.
PlantPax DCS - Template User Manual, Publication 9528-UM001	Provides direction on how to install and deploy PlantPax virtual templates. NOTE: Only needed if you have purchased the PlantPax Virtual Image Templates. <ul style="list-style-type: none"> • Configure the SQL Virtual Server chapter • Configure the Asset Management Virtual Server chapter
PTC ThingWorx Platform 9 Help Center	Provides direction on how to install ThingWorx Foundation for Windows, which Apache Tomcat and Oracle Java installed. PostgreSQL, InfluxDB, or MSSQL Server. NOTE: PlantPax Maintenance Reports were originally tested on the following, however other SQL databases and component versions are available. <ul style="list-style-type: none"> • PostgreSQL v10 • Oracle Java SDK v11 • ThingWorx Foundation v9.2
Vuforia Studio 9.9.0 Help Center	Provides direction on how to install Vuforia Studio.
Vuforia 9.9.0 Experience Service Help Center	The experience service for Vuforia Studio augments.
FactoryTalk Edge Gateway User Manual, publication 95055-UM006	Provides direction on configuring FactoryTalk Edge Gateway.
EtherNet/IP Network Devices User Manual, ENET-UM006	Describes how to configure and use EtherNet/IP devices to communicate on the EtherNet/IP network.
Ethernet Reference Manual, ENET-RM002	Describes basic Ethernet concepts, infrastructure components, and infrastructure features.
System Security Design Guidelines Reference Manual, SECURE-RM001	Provides guidance on how to conduct security assessments, implement Rockwell Automation products in a secure system, harden the control system, manage user access, and dispose of equipment.
UL Standards Listing for Industrial Control Products, publication CMPNTS-SR002	Assists original equipment manufacturers (OEMs) with construction of panels, to help ensure that they conform to the requirements of Underwriters Laboratories.
American Standards, Configurations, and Ratings: Introduction to Motor Circuit Design, publication IC-AT001	Provides an overview of American motor circuit design based on methods that are outlined in the NEC.
Industrial Components Preventive Maintenance, Enclosures, and Contact Ratings Specifications, publication IC-TD002	Provides a quick reference tool for Allen-Bradley industrial automation controls and assemblies.
Safety Guidelines for the Application, Installation, and Maintenance of Solid-state Control, publication SGI-1.1	Designed to harmonize with NEMA Standards Publication No. ICS 1.1-1987 and provides general guidelines for the application, installation, and maintenance of solid-state control in the form of individual devices or packaged assemblies incorporating solid-state components.
Industrial Automation Wiring and Grounding Guidelines, publication 1770-4.1	Provides general guidelines for installing a Rockwell Automation industrial system.
Product Certifications website, rok.auto/certifications .	Provides declarations of conformity, certificates, and other certification details.

You can view or download publications at [rok.auto/literature](#).

Rockwell Automation Support

Use these resources to access support information.

Technical Support Center	Find help with how-to videos, FAQs, chat, user forums, and product notification updates.	rok.auto/support
Knowledgebase	Access Knowledgebase articles.	rok.auto/knowledgebase
Local Technical Support Phone Numbers	Locate the telephone number for your country.	rok.auto/phonesupport
Literature Library	Find installation instructions, manuals, brochures, and technical data publications.	rok.auto/literature
Product Compatibility and Download Center (PCDC)	Download firmware, associated files (such as AOP, EDS, and DTM), and access product release notes.	rok.auto/pcdc

Documentation Feedback

Your comments help us serve your documentation needs better. If you have any suggestions on how to improve our content, complete the form at rok.auto/docfeedback.

Allen-Bradley, CompactLogix, ControlLogix, expanding human possibility, FactoryTalk, FactoryTalk Edge Gateway, PlantPAx, PTC, Rockwell Automation, Sequence Manager, Studio 5000 Logix Designer, ThingWorx, ThingWorx Composer, Visual Studio, and Vuforia are trademarks of Rockwell Automation, Inc.
CIP and EtherNet/IP are a trademark of ODVA, Inc.
Excel, Microsoft, and Windows are trademarks of Microsoft Corporation
Trademarks not belonging to Rockwell Automation are property of their respective companies.

Rockwell Automation maintains current product environmental compliance information on its website at rok.auto/pec.

Rockwell Otomasyon Ticaret A.Ş. Kar Plaza İş Merkezi E Blok Kat:6 34752, İçerenköy, İstanbul, Tel: +90 (216) 5698400 EEE Yönetmeliğine Uygundur

Connect with us.    

rockwellautomation.com — expanding **human possibility**[®]

AMERICAS: Rockwell Automation, 1201 South Second Street, Milwaukee, WI 53204-2496 USA, Tel: (1) 414.382.2000, Fax: (1) 414.382.4444
EUROPE/MIDDLE EAST/AFRICA: Rockwell Automation NV, Pegasus Park, De Kleetlaan 12a, 1831 Diegem, Belgium, Tel: (32) 2 663 0600, Fax: (32) 2 663 0640
ASIA PACIFIC: Rockwell Automation, Level 14, Core F, Cyberport 3, 100 Cyberport Road, Hong Kong, Tel: (852) 2887 4788, Fax: (852) 2508 1846