

A dark blue background with a complex, glowing teal circuit board pattern of lines and nodes.

HighByte

CFE Media
and Technology™

How to make industrial data fit for purpose

Seven steps as a practical guide to data management

Author: John Harrington, HighByte Co-Founder & Chief Business Officer

This article first appeared in its original form in the March 2020 issue of IIoT for Engineers, a CFE Media publication.

Introduction

It seems to me that most of the manufacturers I talk to are drowning in data, and yet they are struggling to make it useful.

A modern industrial facility can easily produce one terabyte of data each day. With a wave of new technologies for artificial intelligence and machine learning—on top of real-time dashboards and augmented reality—we should see huge gains in productivity. Unplanned maintenance of assets and production lines should be a thing of the past.

However, even in 2020, this is not the case. Access to all of this data does not mean it is useful. Industrial data is very raw. The data must be made “fit for purpose” to extract its true value. Also, the tools used to make the data fit for purpose must operate at the scale of an industrial facility.

With these realities in mind, here is a practical, step-by-step guide for manufacturers and other industrial companies to make their industrial data fit for purpose.

Step 1: Start with the use case

Information Technology (IT) and Operations Technology (OT) projects should start with clear use cases and business goals. For many manufacturing companies, projects may focus on machine maintenance, process improvements or product analysis to improve quality or traceability. As part of the use case, company stakeholders should identify the project scope and applicable data that will be required. Make sure the right cross-functional stakeholders are in the room from the project beginning, and that all stakeholders agree to prioritize the project and can reach consensus on the project goals.

Step 2: Identify the target systems

With the use cases and business goals identified, the next step requires identifying the target applications that will be used to accomplish these goals. Characterize the target application by asking these questions:

- Where is this target application located: at the Edge, on-premises, in a data center, in the Cloud, etc.?
- How can this application receive data: MQTT, OPC UA, REST, database load, etc.?
- What information is needed for this application?
- How frequently should the data be updated and what causes the update?

Document your responses and then move on to the next step in which you will identify your data sources for the project.

Selecting Your Integration Architecture

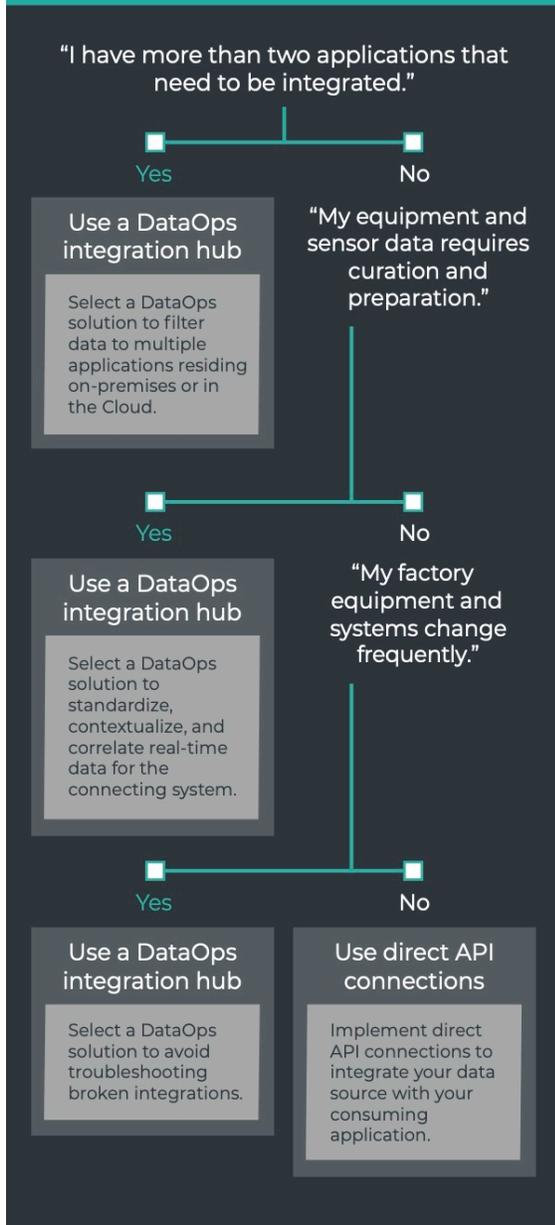


Figure 1. Integration hubs are an alternative to using Application Programming Interfaces (APIs) solely when managing the many types of data associated today with automation projects.

Step 3: Identify the data sources

Industrial data is an important component for addressing industrial and business use cases. However, there are some major challenges with accessing this data and converting it into useful information.

Volume. The typical modern industrial factory has hundreds to thousands of pieces of machinery and equipment constantly creating data. This data is generally aggregated within Programmable Logic Controllers (PLCs), machine controllers or Distributed Control Systems (DCSs) within the automation layer, though newer approaches may also include smart sensors and smart actuators that feed data directly into the software layer.

Correlation. Automation data was primarily put in place to manage, optimize and control the process. The data is correlated for process control and is not correlated for asset maintenance or product quality or traceability purposes.

Context. Data structures on PLCs and machine controllers have minimal descriptive information—if any. In many cases, data points are referenced with cryptic data-point naming schemes or references to memory locations.

Standardization. The automation in a factory evolves over time with machinery and equipment sourced from a wide variety of hardware vendors. This hardware was likely programmed and defined by the vendor. This has resulted in unique data models created for each piece of machinery and a lack of standards across the factory and company in all but the very largest and most sophisticated manufacturers.

You can better understand the specific challenges you will need to overcome for your project by documenting your data sources. Characterize the data available to meet the target system's needs by asking these questions:

- What data is available?
- Where is it located: PLCs, machine controllers, databases, etc.?
- Is it real-time data or informational data (metadata)?
- Is the data currently available in the right format or will it need to be derived?

Step 4: Select the integration architecture

Integration architectures fall in two camps: direct Application Programming Interface (API) connections (application-to-application) or integration hubs (DataOps solutions).

Direct API connections work well if you only have two applications that need to be integrated. The data does not need to be curated or prepared for the receiving application, and the source systems are very static. This is typically successful in environments where the manufacturing company has a single SCADA or MES solution that houses all the information, and there is no need for additional applications to get access to the data.

Direct API connections do not work well when industrial data is needed in multiple applications like SCADA, MES, ERP, IIoT Platform, Analytics, QMS, AMS, cyber threat monitoring systems, various custom databases, dashboards, or spreadsheet applications. Direct API connections also do not work well when there are many data transformations that must occur to prepare the data for the consuming system. These transformations can easily be performed in Python, C#, or any other programming language but they are then “invisible” and hard to maintain.

Finally, direct API connections do not work well when data structures are frequently changing. This happens when the factory equipment or the programs running on this equipment are frequently changed. For example, a manufacturer may have short-run batches that require loading new programs on the PLCs; the products produced may evolve and require changes to the automation; the automation may be changed to improve efficiency; or the equipment may be replaced due to age and performance.

Using the API approach buries the integrations in code. Stakeholders may not even be aware of integrated systems until long after the equipment has been replaced or changes have been made, resulting in undetected bad or missing data for weeks or even months.

An alternative to direct API connections is a DataOps integration hub. DataOps is a new approach to data integration and security that aims to improve data quality and reduce time spent preparing data for use throughout the enterprise. An integration hub acts as an abstraction layer that still uses APIs to connect to other applications

but provides a management, documentation, and governance tool to connect data sources to all the required applications.

An integration hub is purpose-built to move high volumes of data at high speeds with transformations being performed in real time while the data is in motion. Since a DataOps integration hub is an application itself, it provides a platform to identify impact when devices or applications are changed, perform data transformations, and provide visibility to these transformations.

Step 5: Establish secure connections

Now that the project plan is in place, begin system integration by establishing secure connections to the source and target systems. Understand the protocols you will be working with and the security risks and benefits they provide.

Many systems support open protocols to define the connection and communication. Typical open protocols include OPC UA, MQTT, REST, ODBC and AMQP—among others. There also are many closed protocols and vendor-defined APIs for which the application vendor publishes the API protocol documentation. Ask yourself: Does the protocol support secure connections and how are these connections created? Some protocols and systems support certificates exchanged by the applications. Other protocols support usernames and passwords or tokens manually entered into the connecting system or through third-party validation. In addition to user security, some protocols support encrypted data packets so if there is a “man in the middle” attack they cannot read the data being passed. Finally, some protocols support data authentication. Then, even if the data is viewed by a third party, it cannot be changed.

Security is not just about usernames, passwords, encryption, and authentication but also about integration architecture. Protocols like MQTT require only outbound openings in firewalls, which security teams prefer because hackers are unable to exploit the protocol to get on internal networks.

Step 6: Model the data

The corporate-wide deployment and adoption of analytics or IIoT often is delayed by the variability of data coming off the factory floor. From one machine to the next, each industrial device may have its own data model. Historically, vendors, systems integrators, and in-house controls engineers have not focused on creating data standards. They refined the systems and changed the data models over time to suit their needs. This worked for one-off projects, but today’s IIoT projects require more scalability.

The first step in modeling data is to define standard models required in the target system to meet the business goals of the project. At the core of the model is the real-time data coming off the machinery and automation equipment. Most of the real-time data points will map to single-source data points. However, when a specific data point does not exist, data points can be derived by executing expressions or

logic using other data points. Data also can be parsed or extracted from other data fields, or additional sensors can be added to provide required data.

These models also should include attributes for any descriptive data, which are typically not stored in the industrial devices but are very useful when matching data and evaluating data in the target systems. Descriptive data could be the location of the machine, the asset number of the machine, unit of measure, operating ranges or other contextual information. Once the standard models are created, they should be instantiated for each asset, process, and/or product. This is generally a manual task but can be accelerated if the mapping already exists in Excel or other formats, if there is consistency from device to device that can be copied or if a learning algorithm can be applied.

Step 7: Flow the data

When the modeling is complete, the data flows should be controlled model-by-model. This typically is performed by identifying the model to be moved, the target system, and the frequency or trigger for the movement. Over time, data flows also will require monitoring and management.

Wrap up

Factories and other industrial environments change over time. Equipment is replaced, programs are changed, products are redesigned, systems are upgraded, and new users need new information to perform their jobs. Amid this change, OT and IT professionals will collaborate on new projects aimed at improving factory floor productivity, efficiency, and safety. You will need industrial data that's fit for purpose in order to make use of it. And you will need tools to help will be needed to accomplish this task at scale—like a DataOps integration hub. By using an integration hub, administrators can evaluate equipment and system changes, and identify integrations that must be modified or replaced. They can make changes to data models and enable new flows in real time.

Making industrial data fit for purpose will be critical to manufacturers looking to scale their IIoT projects and wrangle data governance. I hope this article serves as a practical guide to getting started on your next project.

About the Author



John Harrington is the Co-Founder & Chief Business Officer of HighByte, focused on defining the company's business and product strategy. John is passionate about delivering technology that improves productivity and safety in manufacturing and industrial environments. He has spent his 25-year career both delivering software to manufacturers and working for manufacturers in operations roles. This experience has given him a unique perspective on how suppliers and end users each play an integral role in implementing new technology solutions. John has a Master in Business Administration from Babson College and a Bachelor of Science in Mechanical Engineering from Worcester Polytechnic Institute.

About HighByte

HighByte is an industrial software development company in Portland, Maine building solutions that address the data architecture and security challenges created by Industry 4.0. We believe contextualized and standardized data is essential for Industry 4.0 to reach broad adoption. That's why we've launched HighByte Intelligence Hub—enabling manufacturers to securely connect, model, and flow valuable industrial data throughout their extended enterprise without writing or maintaining code. HighByte Intelligence Hub is the first DataOps solution purpose-built to meet the unique requirements of industrial assets, products, processes, and systems at the Edge. Learn more and request a free trial at <https://highbyte.com>.