

Upgrade Guide

How to Transition to Ignition by Inductive Automation™

This document will help guide you through the process of upgrading your existing FactorySQL and FactoryPMI projects to Ignition by Inductive Automation.™

Ignition represents a major update to Inductive Automation's product line, and while we've tried to ensure the highest degree of backwards compatibility, some features may not map over one-to-one or may require manual action. In some limited cases, the disparity of features may make it infeasible to upgrade at this time. Inductive Automation will be working to address these issues as quickly as possible.

When to upgrade?

The very first question a user should ask themselves is whether or not they should upgrade to Ignition at this time. Customers with stable running projects which they do not intend to modify should continue using the current version until necessity dictates the upgrade. Similarly, customers with projects that are nearing completion/startup and have been implemented in FactorySQL and FactoryPMI should finish the installation, and not switch over at the last minute. Inductive Automation will continue to support and provide maintenance updates for FactoryPMI and FactorySQL.

Installing Side-by-Side

All customers who are upgrading existing projects are encouraged to install Ignition side-by-side and run both systems concurrently until they are confident that all of the required functionality is present and works as required. Inductive Automation will allow customers to keep both systems activated during this transition period.

It is possible to run FactorySQL, FactoryPMI and Ignition concurrently on a single machine. Whereas FactoryPMI uses port 8080 for communication by default, Ignition defaults to port 8088, avoiding a conflict.

The Upgrade Process

1) Prepare the files

The upgrade process is performed through the Ignition gateway, using system backups from both FactorySQL and FactoryPMI. Of course, users upgrading only one project can ignore references to the other product.

The first step is to gather the appropriate backups. For FactorySQL, a system backup (*.fdb) created from the File>Export menu is the best option. However, a project backup (*.fsp) can also be imported. For FactoryPMI, you'll need a gateway server backup (*.fpgb), created from the System>Backup/Restore menu in the FactoryPMI gateway configuration site.

2) Upgrade the FactoryPMI project

In the Ignition gateway, go to the Configuration section. Log in, and then go to System>Backup/Restore. Select the "Legacy Upgrade" tab. Select your FactoryPMI file, and click "Upgrade" to begin the process.

The FactoryPMI upgrade should be performed first because the process will port over database connections used in the FactoryPMI project. The upgrade tool is currently unable to automatically port database connections from FactorySQL, but since both products are usually configured to access the same databases, running the FactoryPMI upgrade can save you the trouble of needing to create the connections by hand. When the FactorySQL project is upgraded, you'll be given the opportunity to map old database names to the new ones.

3) Create any additional database connections required by FactorySQL

If there are any connections that are required only by FactorySQL (or if you are only upgrading FactorySQL), create them by hand in the Ignition gateway. As mentioned above, the upgrade tool is unable to automatically port SQL connections from FactorySQL. Instead, you will be given the opportunity to map old connection names to the new connections that you create.

4) Upgrade the FactorySQL project

Follow the same procedure outlined in step #2, but for the FactorySQL project. The import will create a new project dedicated to the FactorySQL project. An interesting point of note: whereas FactorySQL only allowed one running project, Ignition allows you to break up groups into multiple projects, and run them concurrently.

5) Configure additional settings from FactorySQL

The FactorySQL upgrade tool currently only imports groups. It will be necessary to configure other features, such as database-based SQLTags and alert notification, by hand in the appropriate sections of the Ignition gateway.

6) Create OPC server connections

FactorySQL did not require explicitly defined connections to OPC servers. Ignition, however, requires that the server connections be created under the OPC Servers section. The OPC-COM module can be used to connect to the same OPC servers that FactorySQL previously used, and has a built-in name resolution feature that will allow the imported groups and SQLTags to find the servers, irregardless of the names you give the connections. A connection should be made to each OPC server used by FactorySQL & SQLTags.

7) Configure additional system settings

Some system features, such as auditing, will need to be reconfigured manually.

Additional Upgrade Possibilities

Ignition includes full support for database-based SQLTags, the methodology used by FactorySQL and FactoryPMI. In fact, it is possible for a legacy FactorySQL to drive SQLTags for an Ignition gateway. Additionally, however, Ignition now supports internal SQLTags – the same SQLTags, but run internally avoiding the overhead of working with the database when it's not required.

The easiest way to transition from database to internal SQLTags is to create a CSV export of your tags from FactoryPMI, and then import them into the core "default" SQLTags provider in Ignition.

Backwards Incompatibilities and Upgrade Caveats

There are a number of cases in which upgrading to Ignition is not recommended at this time. We will continue to improve our upgrade procedure, and will try to address these issues soon. Most of these issues are a result of the dramatic technology shift between FactorySQL and Ignition.

1) FactorySQL redundancy

FactorySQL used database-based failover redundancy, whereas Ignition uses a clustering mechanism similar to FactoryPMI. While this mechanism DOES support redundancy in the SQL Bridge module, some features such as “node ranking” (the ability to specify the desired master machine) do not exist. Users will need to take special care in evaluating the capabilities of Ignition redundancy before deciding to upgrade.

2) Alert Log & Status tables

Ignition has a different log format for the alert log, and no longer stores status information in a database table. There is a new alert querying interface available in scripting that can provide status information, and the history log can also be used to determine current state. Applications that make heavy use of these legacy alerting features will need to be migrated to the new style.

3) FactorySQL Custom Plugins

FactorySQL offered a plugin API that allowed customers to develop their own expression functions. These plugins will not work with Ignition. We hope to soon offer an alternative, but in the mean time projects that rely heavily on these plugins will need to continue to run in FactorySQL.

4) FactorySQL Audio Alerts

This feature does not exist in Ignition. It is possible to play audio in Ignition clients, and there are several discussions in the Inductive Automation Forum that deal with binding audio to events – a subject that can be extended to alert events as well. Applications that use this feature will need to be adapted to use a different mechanism.