Ignition 8 Upgrade Guide

Upgrading from 7 to 8

Introduction

Ignition 8.0 is a major update to the Ignition platform, representing over 18 months of work across virtually all subsystems and modules. Throughout the process, Inductive Automation has strived to maintain its commitment to backwards compatibility and a straightforward upgrade process. However, given the vast scope of changes in 8.0, there are inevitably special considerations that need to be evaluated and actions taken in order to successfully upgrade an existing system. This document seeks to identify those issues and provide guidance for users who are migrating from an Ignition 7 system. To discover the new features of Ignition 8, please see the What's New section.

In some cases, it is not advisable or possible to upgrade from Ignition 7 to 8 at this time. Please read this guide carefully before upgrading a production gateway!

Should I Upgrade?

It may not be advisable for all users to immediately upgrade production systems to 8.0. It is highly recommended that all users who wish to upgrade should first install Ignition 8 on a separate computer and validate all of the functionality required by their system before deploying it into production. Please read this guide in its entirety to understand the changes introduced in Ignition 8 and evaluate if they will impact your installation(s).

Expected Downtime

If you are upgrading from Ignition 7.9 or before to Ignition 8, there are some changes that may require a downtime window, even when using Ignition’s Redundancy. These changes are covered in the rest of the upgrade guide. For example, for anyone using Vision, all Vision Clients will need to be updated to run using the new Vision Client Launcher, which will need to be installed on each client PC.

Changes to the Release Cycle

Updates to Ignition 8 will be made according to a new release cadence that targets smaller, more frequent, and more predictable releases. The numbering methodology will continue to be the same as before (eg. 8.0.1, 8.0.2), but will now occur at regular intervals. At any given time, multiple branches will be available on the downloads page: Stable, RC, and Nightly Beta.

As part of this effort, we will also be improving the Change Log section of the website, and will strive to provide more information to users in regards to upcoming changes and known issues.

Please see the More Information section at the end of this document for links to various resources where update information will be published.

Using Perspective without Upgrading

The Ignition Perspective module is one of the most exciting new features of Ignition 8, offering an entirely new and innovative way to visualize data and deploy applications to mobile devices. Many users will want to begin leveraging this module right away, and will feel compelled to upgrade for that purpose.

It is important to understand how various Ignition features, in particular the gateway network and remote services, can be leveraged to allow immediate usage of Ignition Perspective without the potential risk or disruption necessary to upgrade a running production gateway.

As part of the base platform, the Gateway Network allows you to create connections between gateways, including between 8 and 7. The Remote Tag Provider allows you to use tags from one gateway on another, including both realtime and historical values. Remote Tag Providers created in...
Ignition 8 targeting Ignition 7 gateways will support reading, writing, real-time subscriptions, and tag history queries. You cannot edit or modify the tags.

Therefore, to develop and deploy Perspective applications with minimum impact on your production system you can:

1. Install Ignition 8 onto a different computer or VM instance.
2. Limit the installation to the Perspective module (though, of course, other modules can be used)
3. Create a gateway network connection between the Ignition 8 gateway and the Ignition 7 gateway.
4. Create a Remote Tag Provider on the Ignition 8 system for each provider you’d like to access.

Unless previously changed, remember that the Ignition 7 gateway will employ the default “service security” settings for the remote tag gateway network service. This will allow read only data, but if you want to support writes, you’ll need to change the policy by going to Configure>Security/Service Security in the gateway.

Important Changes

This section describes four levels of changes that users should be aware of when upgrading a system:

- Discontinued features
- Currently incompatible changes
- Changes that require user action
- Changes to be aware of

Discontinued Features

These are areas that have been removed or discontinued, as new methodologies have been adopted to replace the functionality.

Vision Mobile Module Discontinued

The Vision Mobile Module is no longer supported in Ignition 8. The new Ignition Perspective Module provides a robust, modern solution for mobile visualization. Users should migrate their mobile visualization needs to be based on Perspective applications going forward.

A few important points:

- The Mobile Module will continue to be supported on Ignition 7.7 and 7.9 for the remainder of their respective Long Term Support (LTS) terms.
- All customers who own the Mobile Module and have current TotalCare Upgrade Protection contracts are entitled to a license of
Perspective Mobile upon upgrade.

External Database “SQLTags” Providers No Longer Supported

The “Database Provider” and “Database Driving Providers” are no longer supported in Ignition 8. These systems were very early ways to bridge outside data into Ignition’s tag system and have been replaced over the years with much easier and faster methods of data integration. If you are still using these features, please contact us to explore migration strategies.

32 bit Support

Support for 32-bit platforms other than ARM has been discontinued.

Project Rollback

The Project Rollback feature has been discontinued in Ignition 8.

Currently Incompatible Changes

These are areas that are not yet fully available or compatible in 8.0.0, but that we expect to progress soon. Please see the references in “More Information” to track how these develop.

Sepasoft Modules

Please follow this link to Sepasoft’s website for information about module compatibility with Ignition 8. https://help.sepasoft.com/docs/display/MHD/Ignition+8+Readiness

Cirrus Link Modules

Please follow this link to Cirrus Link’s website for information about module compatibility with Ignition 8. https://docs.chariot.io/display/CLD/Ignition+8+Status+Updates

3rd-party Modules

Please consult individual module authors concerning upgraded module availability. Modules built for Ignition 7 will not work in Ignition 8 without modification.

Staging & Publishing

Ignition 8 features a completely revamped project system. This new project system does not have different “staging” and “published” versions of each project. When you upgrade from Ignition 7, any projects that were split into separate staging and publish versions will have their staging versions imported into Ignition 8. This is to prevent any loss: the staging version is always at least as and usually more up-to-date than the published project.

Staging and publishing was not included in Ignition 8’s project subsystem for two reasons:

1. Staging & Publishing was not consistently implemented across various modules, leading to confusion about which types of resources participated in it.
2. Projects are not self-contained: they rely on tags and external connections to devices, databases, etc. For this reason, the staging & publishing feature was only effective in limited circumstances.

We plan on bringing back the idea of staging and publishing in the near future, except that it will be a more comprehensive approach that will affect entire gateways. Many users already use multiple gateways to create separate environments: this is a concept that we can make more powerful by building support for it into the platform. In the meantime, you can emulate the old staging & publishing system by simply having a “staging” copy of your project and either move resources between projects or copy & overwrite them in order to “publish” your changes.

OEM Lock

The “OEM-Lock” feature that was used to encrypt project resources by authors is not currently supported in 8.0. An updated version of this feature will be introduced at some point. Project authors that used this feature may migrate their projects to 8.0, as the owner license will decrypt Ignition 7 resources, but they should be aware that the resources will currently remain unencrypted.
End users who upgrade OEM-Locked systems to 8.0 will not be able to view the project resources.

**OPC UA Server Exposed Tags Auditing**

When a tag made available via the “Exposed Tags” feature of Ignition’s OPC UA server is written to by an OPC UA client the write is not currently entered into the audit log. This will be re-implemented in a future version.

**OPC UA Server “ReadOnly” Role**

The undocumented “ReadOnly” role no longer has any effect. First class support for role-based access control will be implemented in a future version.

**Changes That Require User Action**

These changes will require manual action on the part of the system administrator/project author. In other words, these are features or situations that have a direct corresponding solution, but for various reasons cannot be automatically upgraded.

**Parameter References in Tag Event Scripts**

UDT Parameters have changed dramatically in Ignition 8 from a technical point of view (see “Changes to Be Aware Of” below for more information). In most cases, the system will update the references automatically. However, usage of parameters in Tag Event Scripts must be manually modified.

**Ignition 7**

Parameter references were replaced in a “pre-compilation” phase before building the executable Tag. This means that all references were directly replaced with their literal values before compiling the Tag Event Script.

Example:

```
myIntParam = {MyIntParam}
myStringParam = “{MyStringParam}”
```

**Ignition 8**

Parameters are now considered properties of a Tag, inherited from the containing data type. There is no longer the notion of “pre-compilation”, but instead dynamic bindings are created where appropriate, or in the case of event scripts, the properties may be accessed through the new “Tag” parameter.

Example:

```
myIntParam = tag[‘MyIntParam’]
myStringParam = tag[‘MyStringParam’]
```

**Redundancy Connection Configuration**

Redundancy communication is now accomplished through a Gateway Network connection between the master gateway and the backup. This mechanism provides greater security and performance than the previous communication channel, but will require the system administrator to re-establish the connection.

The settings for the communication channel are still configured in the gateway under Configure>Redundancy, and are still established from the Backup to the Master, but must be modified before the backup will successfully connect to the master after upgrade.

**Changes to Be Aware Of**
Ignition 8 has many additional changes and improvements that system administrators should be aware of. Although none of them present a significant issue, some may directly affect end users (for example, launching clients), and some can be leveraged to improve the security and performance of your system.

Introducing Ignition Perspective

Ignition Perspective is an entirely new visualization system, built from the ground up to allow you to create modern, high performance, and highly functional pure-web applications that can be viewed anywhere. Learn more about Perspective by visiting the Perspective pages in the User Manual, or the Inductive University.

System Commissioning

Upon installing Ignition, the user is now presented with a “system commissioning” web based walkthrough. This process must be completed for new installations, and requires that the administrator create a root username and password combination, specify the ports that Ignition will use, and accept the appropriate licenses.

As of Ignition 8.0, there is no longer a pre-defined default “admin” user. System administrators should choose a secure root username and password for optimal system security.

Java Web Start Discontinued

Support for Java Web Start has been discontinued by Oracle and the OpenJDK community as of Java 11. This technology has been used in Ignition since its original release to launch the Designer and Vision clients, though we have also offered alternative launching tools for years.

In anticipation of this change, the designer and client launching tools have been greatly improved in Ignition 8. In addition to a fresh and more functional UI, the launch tools now manage the Java runtime environment used by the client (see next point), and can help you track many projects and gateways. There are separate launchers for Vision Clients and the Designer, representing the natural distinction between the target users of these products, and the launchers can be downloaded for Windows, Mac, and Linux directly from the gateway homepage.

Java Web Start is no longer supported, please use the Designer and Client launchers to launch those applications.

Embedded Java

Ignition no longer requires Java to be installed on the gateway or client machines. Native versions of the Ignition installer and launchers have been made available for 64-bit Windows, Mac, and Linux, as well as ARM, for use in edge devices. These distributions include embedded JREs (Java Runtime Environment) that are provided by Inductive Automation and managed by Ignition.

The native client launchers manage retrieving the current version of the JRE as necessary for the client and designer runtimes. Updates to the embedded JREs are managed by Inductive Automation and are made available through updates to Ignition. The client launchers will automatically update their local versions as necessary, ensuring that Vision Clients and the Designer are always running the latest version available on the Gateway.

Ignition no longer requires Java to be installed on server or client machines. System administrators should monitor Inductive Automation for important system and security updates, and can sign up for important notifications at http://support.inductiveautomation.com

Project Inheritance & “global” upgrade logic

The project system in Ignition 8 has been improved and now features a more flexible inheritance model. Each project may have a “parent” project, and will inherit all of the resources of that parent project. That project may in turn have its own parent project, and so in this way complex hierarchies of re-usable resources may be designed. Within each project, inherited resources may be used by other, “local” resources. For example, an inherited Vision template could be embedded in a window, or an inherited script could be executed by a button. Inherited resources may also be overridden, allowing them to be re-defined when needed. Conversely, resources in parent projects may be marked as “not overridable” to ensure that children cannot alter them, when appropriate.

This new inheritance system makes Ignition 7’s “global” project concept obsolete. In order to maintain backwards compatibility, the following will happen when an Ignition 7 Gateway is upgraded to Ignition 8:

- The resources from the Ignition 7 special “[global]” project will be copied into an Ignition 8 standard, inheritable project called “global”.
- All other projects will have their parent project set to this new “global” project.
- All “shared” scripts are now simply project-level scripts in a top-level script package called “shared”, stored in the new “global” project. This allows all existing scripts to function without edits.
- All “project” scripts are now simply project-level scripts in a top-level script package called “project”.
- All Vision templates stored in the old global area are moved into a folder called “SharedTemplates/” in the new global project. The vision module will automatically convert a template path like “[shared]MyTemplate” as “SharedTemplates/MyTemplate” so that no further action is needed.
- If any run-always sequential function charts are present in the global area, a new project called “run-always-charts” will automatically
be created upon upgrade, and these charts will be moved into this project. This is because resource types that execute automatically like run always SFCs and Transaction Groups do not run in inheritable projects.

- If any alarm pipelines exist, a project called “alarm-pipelines” will be created and all alarm pipelines will be moved into this project. Tags can target pipelines in any project, but in the absence of a specific project, the pipeline is assumed to be in this special upgrade project.
- There is a new gateway-wide setting called “Gateway Scripting Project”. This setting lets you specify one project which all scripts that are outside the scope of a project will execute against. For example: Tag Change Scripts and Tag expressions using the runScript expression. On upgrade, this will be set to the new “global” project so that Tag scripts can continue to access shared.* scripting functions.

Project Resource Storage

The new project subsystem in Ignition 8 uses a new file-based storage strategy, instead of the internal-database strategy used in Ignition 7. This has several significant benefits over the previous methodology:

1. All resources can now be tracked with industry standard source control tools, such as Git. Resources can be tracked individually, and changes to disk caused by source control operations (checkout, pull, etc) will be applied automatically to the running system.
2. Project resources can be transferred between systems with simple file system operations. System administrators can use a wide range of tools to synchronize all types of resources between Gateways.
3. Project backups are simply standard zip files organized in the same way as the file system.

Designer Concurrent Editing and Conflict Resolution

Ignition 8’s Designer uses a new lock-free strategy for handling concurrent editing. We believe that this strategy is significantly more robust and will allow for more designers to work together on projects.

The Ignition 7 locking strategy (besides being frustrating when users would lock resources and then go home for the day!) is incompatible with the new filesystem storage strategy. Since changes can occur in the filesystem at any time, it isn’t possible to maintain exclusive edit locks in the Designer.

In Ignition 8, you may open and modify any resource in your project. You will be notified if the system detects that you and another user are both editing the same resource, or if the resource you’re editing has already been modified remotely. If you modify the resource anyway, and someone else has modified it first, this is called a conflict. The system automatically detects conflicts when you try to save. If any are detected, you are given the opportunity to resolve these conflicts by choosing whether to use your changes, accept the changes that happened elsewhere, or cancel your save and figure out what to do in another way.

Tag System Changes

The Tag system has undergone significant changes in Ignition 8. The majority of these changes should go unnoticed upon upgrade (apart from obvious UI changes), and most systems should immediately benefit from improved performance, faster edits, and expanded functionality. Under the hood, however, many technical aspects have changed. Understanding these changes are important for design and troubleshooting in Ignition 8.

UDT Performance

The tag system has been redesigned to optimize for UDT oriented design. In Ignition 7, certain aspects of UDT performance led some system designers to favor standard Tags over UDTs for some applications. Such systems should be re-evaluated in Ignition 8, as use of User Defined Types will lead to much more efficient memory usage and lighter weight change processing.

UDT Parameter Usage

As mentioned in the section Changes That Require User Action, the way that Parameter references are applied to Tags has changed. Instead of being replaced in a “pre-compilation” stage of Tag execution, they are now treated as true property references. This means that they can be modified, and those modifications propagate dynamically to referencing properties, causing those to be updated in turn.

An important consequence of these changes is that syntax for parameter references in Expressions has changed in a subtle, but crucial way. In Ignition 7, due to the replacement strategy, the following would have been a valid expression:

```java
"The value of the parameter is: {ParamRef}"```

In Ignition 8, this value would no longer be a valid reference. It is counter-intuitive to all other uses of reference in Ignition, and has been replaced with a more familiar syntax:

```java
concat("The value of the parameter is: ", [ParamRef])```
All existing references will be upgraded automatically to the new syntax upon system upgrade. Designers should be aware of this change when creating new Tags.

Script Based Editing

The various scripting functions for editing Tags (system.tag.editTag(s), addTags) have been deprecated and replaced by system.tag.configure(). The previous functions will continue to work, but designers should learn the new function, which is far more intuitive and powerful than the previous functions.

Tags are now defined as JSON objects, which consist of properties, arrays, and sub-objects. The system.tag.configure function can take either a String document definition, or a JSON object that defines one or more Tags. Overrides for UDTs are created by simple redefinition of properties, and complex structures like Event Scripts and Alarm configurations will be merged with inherited definitions.

OPC UA Certificate Management

Prior to Ignition 8, Ignition’s built-in OPC UA client implicitly trusted the certificate of any server it connected to, and Ignition’s built-in OPC UA server implicitly trusted the certificate of any client connecting to it. This is no longer true. New certificate management pages for the client and server have been added to the configuration section of the Gateway under “OPC UA > Security”. From this UI, trusted certificates can be imported and quarantined certificates can be marked as trusted. Ensuring the remote certificate is trusted is required for all secured inbound and outbound connections.

On upgrade, all secured inbound and outbound connections other than the default “loopback” connection will be faulted until the remote certificate is explicitly marked as trusted.

OPC UA Default Configuration Changes

New installations have some new OPC UA related defaults:

- The server binds only to localhost (i.e. no remote connections allowed)
- The server uses port 62541 instead of 4096
- The server only allows secured connections (SecurityPolicy Basic256Sha256)
- The default “loopback” connection is called “Ignition OPC UA Server” (there is no longer a “-” between “OPC” and “UA”)

Upgrades are not affected by these changes; they retain their existing configuration.

Look and Feel

The “look and feel” of the Ignition Designer and Vision Client has been updated. This may result in subtle changes to the appearance of Vision Clients depending on the components used by a given project. It is not possible to revert to the old “look and feel” as it’s not compatible with Java 11.

More Information

To learn more and to stay up to date on updates to Ignition 8, check out the following resources:

- Inductive Automation’s website: many product pages have been updated with information about Ignition 8 and the Perspective Module.
  - The Blog has new information concerning Ignition 8 coming over the next few months.
  - The downloads page has been modified, and will offer new nightly builds of the latest Ignition 8 updates, in addition to stable and RC versions.
- support.inductiveautomation.com
- Forum.inductiveautomation.com - The online forum is a great place to provide feedback and to learn about what other users are encountering with Ignition 8.
- Inductive University has many new videos covering a wide range of Ignition 8 topics. New videos will also be released regularly.
- This User Manual is being continually updated with new content as Ignition evolves.

Related Topics ...

- Installing and Upgrading
- Installing or Upgrading a Module