

Ignition 8 Upgrade Guide

Should I Upgrade?

It may not be advisable for all users to immediately upgrade production systems to 8. It is highly recommended that all users who wish to upgrade should first install Ignition 8 on a separate server 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.

Upgrading from 8.0 to 8.1

This section details notes and advisories for users planning on upgrading from 8.0 to 8.1.

Changes to Be Aware Of

Tag Historian Licensing

In 8.1, the Tag Historian module licensing has been adjusted.

1. Licensing for the full Tag History works identically to before. If you have a normal Tag Historian license, and are using the standard Tag Historian functionality, there are no changes.
2. For customers with a Scale-Out architecture, the Tag Historian could be used on the front-end Gateways unlicensed. This would provide a message of "Partially licensed" on those Gateways, even though the configuration was an acceptable architecture with no need for additional licensing. In 8.1, we added the ability for the Tag Historian to be licensed for read-only operations, to eliminate that message. This is available for free for users upgrading from existing licenses 7.9 and 8.0 licenses that have Upgrade Protection by contacting your sales representative at Inductive Automation.
3. A small set of customers use third-party modules that act as a storage engine for Ignition's Tag Historian. Previously, Inductive Automation did not require a Tag Historian license for a module to act as an engine for our Tag Historian (i.e. to use our API for storage and retrieval). Starting in 8.1, a Tag Historian license is needed in order for these modules to function. If you are using one of these modules in 7.9 or 8.0 and you have Upgrade Protection on your license, Inductive Automation will provide a free addition to your license, so no additional cost is required upon upgrade. If you have Upgrade Protection, Inductive Automation will attempt to auto detect if you're running one of these modules to update your license in our system automatically, so you would only need to click the refresh license button in your Ignition Gateway to pull in the change. Some third party modules cannot be auto detected, so call your sales representative if a third party historian module does not work on upgrade and you have an existing 7.9 or 8.0 license with Upgrade Protection, and we will add the necessary parameters to your license.

If your system is one of the few that uses third-party historian engine modules, it is highly advised that you review the system after upgrading to 8.1 and ensure the historian system is working as expected. If there are any issues with historian system after upgrade, and clicking refresh on your license hasn't resolved the issue, please contact your sales representative for assistance.

Gateway Scan of Project Directory

As of 8.1, the Gateway scans the project directory every **five minutes** for changes. This is an important change for users that have a Version Control System with their Gateway. Previously in 8.0, the scan time was every **10 seconds**, which proved to be resource intensive.

Users who want to change this scan rate back to the 10 second frequency can do so by placing the following system flag in the ignition.conf file:

```
-Dignition.projects.scanFrequency=10
```

JxBrowser Library Update

On this page ...

- [Should I Upgrade?](#)
- [Expected Downtime](#)

Upgrading from 8.0 to 8.1

- [Changes to Be Aware Of](#)
 - [Tag Historian Licensing](#)
 - [Gateway Scan of Project Directory](#)
 - [JxBrowser Library Update](#)
 - [Gateway Authentication with Identity Providers](#)
 - [Perspective Script Transforms](#)

We updated our bundled JxBrowser library from version 6 to version 7. For most users, this has no ramifications at all.

For users that have intentionally written scripts against the JxBrowser library, those scripts may have broken as a result of the update. Take a look at the [JxBrowser migration guide](#) for more details.

Gateway Authentication with Identity Providers

In Ignition 8.1, Identity Providers are used to authenticate users on the gateway, whereas version 8.0 and prior used User Sources. To make the upgrade process seamless, your newly upgraded 8.1 gateway will create a new Ignition Identity Provider, and point it to the System User Source (under Gateway Settings prior to upgrade). For this to work, the Ignition Identity Provider needs to be able to view the list of users from the backing User Source.

In cases where Active Directory or Hybrid User Sources were set as the System User Source, you'll want to make sure that the list of users from the underlying user source are available. Otherwise, some users may not be able to log into the gateway after upgrading. You can validate this before upgrading by viewing the [Manage Users](#) page. If one or more users are missing from the resulting list, then you'll want to refine how the user source builds that list. Based on the system's user source, you may need to check some settings:

- For Active Directory User Sources, it's a good idea to verify the **User Search** and **User List filter** settings on the User Source, making sure they're filtering on the criteria you expect.
- For Hybrid User Sources, the **List Users from Active Directory** will determine where the list of users is generated from. If the resulting user list is missing users, you may need to toggle this setting.
- For Database User Sources using the "Manual" Mode, make sure the **List Users Query** is able to return the users you'd expect.

If, after upgrading, you're unable to log into the Gateway you can always perform a [gateway password reset](#) to troubleshoot any authentication issues.

Perspective Script Transforms

Arrays in script transforms will now automatically be returned as valid JSON. This change technically modifies the value returned by the transform (i.e., potentially adding quotation marks around string elements), which may impact some existing bindings. If you're making use of Script Transforms that interact with arrays, then you may want to reexamine their configuration after upgrade.

Upgrading from Ignition 7 to 8.1

- [Upgrade Process](#)
 - [Using the Installer to Upgrade \(Preferred Approach\)](#)
 - [Importing Projects into a Fresh Install](#)
- [Changes That Require User Action](#)
 - [Look and Feel changes](#)
 - [Redundancy Connection Configuration](#)
 - [Replacement for the system.tag.browseConfiguration Function](#)
 - [OPC UA Certificate Management](#)
- [Discontinued Features](#)
 - [Vision Mobile Module Discontinued](#)
 - [External Database "SQLTags" Providers No Longer Supported](#)
 - [32 bit Support](#)
 - [Project Rollback](#)
 - [OPC UA Server Stale Threshold](#)
 - [Staging and Publishing](#)
 - [OEM Lock](#)
 - [OPC UA Server Exposed Tags Auditing](#)
 - [OPC UA Server "ReadOnly" Role](#)
- [Changes to Be Aware Of](#)
 - [Third-Party Modules](#)
 - [Tag Historian Licensing](#)
 - [Default Security Certificate](#)
 - [Native Client Launchers Replaced](#)
 - [Introducing Ignition Perspective](#)
 - [Introducing Identity Providers](#)
 - [System Commissioning](#)
 - [Security Levels](#)
 - [Expanded Auditing Scope](#)
 - [Java Web Start Discontinued](#)
 - [Embedded Java](#)
 - [Project Inheritance & "Global" Upgrade Logic](#)
 - [Project Resource Storage](#)
 - [Designer Concurrent Editing and Conflict Resolution](#)
 - [Tag System Changes](#)
 - [UDT Performance](#)
 - [UDT Parameter Usage](#)
 - [Script Based Tag Editing](#)
 - [OPC UA Default Configuration Changes](#)
 - [OPC UA Discovery with Third-Party Clients](#)
 - [Changes to system.alarm.listPipelines](#)
 - [SQLite Storage](#)
- [More Information](#)

Upgrading from Ignition 7 to 8.1

Ignition 8.1 is a major update across virtually all subsystems and modules. Inductive Automation strives to maintain its commitment to backwards compatibility and a straight-forward upgrade process. However, given the vast scope of changes in 8.1, 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 8.1, please see the [What's New](#) section.

Note: In some cases, it is not advisable or possible to upgrade from Ignition 7 to 8.1 at this time. Please read this guide carefully before upgrading a production Gateway!

Upgrade Process

When upgrading from Ignition 7 to Ignition 8.1, there are two general approaches.

Using the Installer to Upgrade (Preferred Approach)

Using the Installer to upgrade is the easiest approach. Simply run an Ignition 8 installer on your existing server (similar to how upgrades normally work). As always, we highly advise taking a Gateway backup before upgrading.

Importing Projects into a Fresh Install

We have found that some users prefer to completely uninstall their Ignition 7 Gateways, install a fresh copy of Ignition 8.1, and then manually import projects and Tags from the old Ignition 7 Gateway, one-by-one. We generally **discourage** this practice, as it results in unintended results. We recommend you take the "Using the Installer to Upgrade" approach instead. However, if you must import Ignition 7 resources into an Ignition 8.1 Gateway, then we recommend you use Gateway Backups instead (when possible).

The reason for this, is that a Gateway Backup triggers an upgrade check that looks over the Gateway Backup and makes modifications to the resources so they run as expected on an Ignition 8.1 system. The same check is triggered when running the installer. Project and Tag imports **do not** trigger this check. As a result, you may need to make manual modifications to the imported resources.

For example, the project system in Ignition 8.1 is drastically different, so running the installer directly on an Ignition 7 Gateway or importing an Ignition 7 Gateway Backup into an Ignition 8.1 Gateway will trigger all of the updates listed under the [Project Inheritance & "Global" Upgrade Logic](#) heading below. Individually importing projects one-by-one will not trigger the upgrade logic, so any legacy "shared" scripts may not work in Ignition 8 without user intervention.

The same could be said for Tag imports. Ignition's built-in OPC UA server connection on Ignition 7 used the name "Ignition OPC-UA Server", but Ignition 8.1 changed the name to "Ignition OPC UA Server", removing the hyphen "-" character. As a result, your Ignition 7 Tags are likely using the old name, where as a freshly installed Gateway is using the new name. Thus, OPC Tags imported from an Ignition 7 file will likely break upon import as the OPC Server name won't match, so you'll need to either A) rename the OPC UA server connection or B) find-and-replace the OPC UA server name on all of the Tags.

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.

Look and Feel changes

The general 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.

This means screens that have been developed in version 7 may look different after upgrade. Notably, the default font on many Vision components is larger, so you may need to resize the font to match your desired look. Manual configurations to Vision resources, such as manually selecting a font or background color for a component, will be preserved upon upgrade. However, visual properties left with default values may appear differently after upgrade.

Note: It is highly advised to upgrade version 7 projects to version 8.1 on a test server and verify the look of the project, before upgrading the production servers.

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.

Replacement for the `system.tag.browseConfiguration` Function

The Tags system in Ignition changed dramatically in Ignition 8, which makes the Ignition 7 function `system.tag.browseConfiguration` obsolete. The function has been removed from the code base, so scripts using the old `browseConfiguration` function will break upon upgrade. The functionality has been replaced by [system.tag.getConfiguration](#).

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.

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 7.9 for the remainder of its 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.

OPC UA Server Stale Threshold

The Stale Threshold setting on Ignition's OPC UA server has been retired. The setting was originally used in earlier versions of Ignition as a means to help some of our drivers determine when values should be marked as stale. Our drivers have since gaining their own methods to determine staleness. The setting has little use in Ignition 8, so it was removed.

Staging and 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 and 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 and publishing feature was only effective in limited circumstances.

You can emulate the old staging and 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 Ignition 8. 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.

OPC UA Server “ReadOnly” Role

The undocumented “ReadOnly” role no longer has any effect.

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.

Third-Party Modules

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

Tag Historian Licensing

See the Tag History Licensing changes under the [Upgrading from 8.0 to 8.1](#) section.

Default Security Certificate

Ignition installations no long come with initial security certificates.

Native Client Launchers Replaced

The Native Client Launcher has been replaced by the [Designer Launcher](#) and [Vision Client Launcher](#), which offer similar functionality. Due to the aforementioned "Default Security Certificate" change, secure connections between the launchers and Gateway will require valid certificates are in place.

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](#).

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.

Introducing Identity Providers

Ignition 8 adds support for connecting to SAML and OpenID Connect Identity Providers. In addition, Ignition can act as an Identity Provider, using a User Source as a backing source of users. On upgrade, a new Ignition Identity Provider will be created that points to a single user source, allowing users in that user source to authenticate against the Identity Provider. You can learn more about this on the [Configuring Identity Providers](#) page.

For more information, see the [Gateway Authentication with Identity Providers](#) under the Upgrading from 8.0 to 8.1 section.

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 an initial user is created for the newly commissioned Gateway.

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.

Security Levels

Ignition 8 introduces the concept of [Security Levels](#), which are abstracted attributes that can be given to users, and used as a means to restrict access to different parts of the system. Security Levels can be assigned to users based on User Roles, Security Zones, a combination of both, or neither, as the Expression Language can be used to determine which level any given user falls under.

Many subsystems in Ignition will automatically migrate over to security levels upon upgrade, such as Tag read and write permissions.

Expanded Auditing Scope

Ignition's built-in auditing system was very limited in version 7, recording only a handful of events. Most events that registered in the system had to originate from a project, so changes made at the Gateway level were largely ignored, save for Tag writes.

In Ignition 8, the auditing system's reach has broadened greatly. The system now accounts for configuration changes made at the Gateway level. Check out the [Auditing Actions Reference](#) page for a full listing of actions that are audited.

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 file system storage strategy. Since changes can occur in the file system 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. To summarize:

- "Scan Classes" have been renamed to "Tag Groups". Functionally they're identical. Scan Classes on Ignition 7 systems will gracefully migrate to Tag Groups upon upgrade.
- The Tag Editor has been gone through a major UI rework, but remains as easy-to-use as ever.
- Tags are fundamentally defined via JSON structure now. A new right-click menu has been added to the Tag Browser, allowing you to edit the raw JSON of a given Tag. Additionally, Tag exports/import files now support a JSON format.

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:

```
"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:

```
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 Tag Editing

The various scripting functions for editing Tags (`system.tag.editTag`, `system.tag.addTag`, etc.) 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 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.

OPC UA Discovery with Third-Party Clients

Ignition's OPC UA server disallows anonymous access on new installations, which can make the server endpoint difficult to discover by UA clients. To help with this, Ignition's UA server runs a separate unsecured endpoint to allow discovery. The discovery endpoint is accessible by appending `/discover` to the end of the URL:

```
opc.tcp://1.2.3.4:62541/discovery
```

Changes to `system.alarm.listPipelines`

The `listPipelines` function was changed to search a single project for pipelines. Upgrading from Ignition 7 to Ignition 8 will migrate all alarm pipelines into a project named "alarm-pipelines". To maintain backwards comparability, `system.alarm.listPipelines()` will check for a project named "alarm-pipelines", but now accepts a project name as a parameter, allowing the function to check for pipelines in any project.

SQLite Storage

Gateways can now create SQLite databases, or "Internal databases" for many of the storage subsystems (Database Connections, Historian, Alarm Journal, and Auditing), allowing your gateway to store and retrieve data without needing to connect to a remote SQL database, which is useful for development systems and demonstrations.

Note: We still recommend external SQL databases are used in production systems, as they generally offer a level of performance and redundancy not available to SQLite databases.

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 updates to Ignition 8.
 - 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](https://www.inductiveautomation.com/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. You can check out the updates for each new version on the [New in this Version](#) page.

Related Topics ...

- [Installing and Upgrading](#)