

# Ignition 7.9 to 8.1

## Upgrade and Migration Procedure



# Overview

This procedure is used to describe the migration process from Ignition 7.9x to Ignition 8.1x. It includes details on the overall best practices for upgrades, information on upgrading Ignition projects, and the steps to deploy the upgrade to production.

## Recommended Process Overview

1. Create an additional VM to be used as the **Upgrade VM**
2. Clone the existing **Production** solution to the **Upgrade VM**
3. Upgrade the **Upgrade VM** from 7.9.x to 8.1.x
4. Validate application functionality and make corrections as necessary on the **Upgrade VM**
5. Deploy to production – approximately 2-hour system downtime
  - 5.1 Take Ignition/Database backups of **Production**
  - 5.2 Uninstall Ignition 7.9.x
  - 5.3 Install Ignition 8.1.x
  - 5.4 Migrate solution from **Upgrade VM** to **Production** (using Ignition Gateway Backup)
  - 5.5 Replace 7.9 Vision client launchers with 8.1.x Vision client launchers
  - 5.6 Validate **Production** functionality acts as expected
6. The **Upgrade VM** is no longer required and can be decommissioned.

## Best Practices for Upgrades

1. The upgrade should be completed in isolation. Do not change features within the system in parallel with the upgrade.
2. If the Sepasoft modules are utilized on the gateway and a move from Sepasoft 2.0 to Sepasoft 3.0 is planned, upgrade Sepasoft to 3.0 before upgrading Ignition from 7.9 to 8.1.
3. Take advantage of Find/Replace to accelerate the upgrade process

# Upgrading the Ignition Projects

1. Take backups of the 7.9.x system
  - 1.1 This includes any databases that are need for proper operation of the Ignition application in a DEV environment
  - 1.2 This also includes a Gateway backup
2. Install the ignition 7.9.x version on a DEV environment
3. Restore the backups that previously took on the DEV environment
  - 3.1 This may require installing a Database server instance in the DEV environment as well to recreate the Database connections with Ignition in the DEV environment.
  - 3.2 The goal is for the DEV environment to match the Production environment as much as possible.
4. Download and install the version of Ignition you will be upgrading to (8.1.x) on the DEV environment
  - 4.1 During this, it will recognize that a previous version is installed and will ask if you want to proceed with the upgrade. Select “Yes” if asked.
  - 4.2 The reason for doing the upgrade, instead of importing 7.9.x project files into an 8.1.x Gateway, is the upgrade process triggers an upgrade check that looks over the Gateway Backup and makes modifications to the resources. This is so they run as expected on an Ignition 8.1 system.
5. When the upgrade is complete, there are functions that are deprecated in the new version of Ignition that will need to be changed.
  - 5.1 These are the most notably the **system.tag.browseTags()**, **system.tag.read()** and **system.tag.write()**  
**browseTags() has been replaced by browse()**  
The params have been changed slightly, so find all locations where it is being used and make changes according to Ignition’s documentation on the new function (using “Find/Replace” is helpful).  
**read() has been replaced by readBlocking()**  
The params have been changed slightly to accept a list of tag paths. Find all locations where it is being used and make changes according to Ignition’s documentation on the new function (using “Find/Replace” is helpful).

**write() has been replaced by writeBlocking()**

The params have been changed slightly to accept a list of tag paths. Find all locations where it is being used and make changes according to Ignition's documentation on the new function (using "Find/Replace" is helpful).

6. There are changes to the way UDT parameters are used, so go through UDT's and check for the following:
  - 6.1 UDT params are now treated as true property references
    - 6.1.1 The syntax has changed. See the examples below:

**7.9.x:**  
"The value of the parameter is: {ParamRef}"

**8.1.x:**  
concat("The value of the parameter is: ", {ParamRef})
  - 6.2 Quotes are taken literally in string UDT parameters
    - 6.2.1 String UDT parameters that contained quotation marks in 7.9 will now be represented literally. In 7.9, the quotation marks were omitted. After upgrading, the quotation marks will be included in the UDT parameter value.
  - 6.3 The instanceName UDT Parameter works differently
    - 6.3.1 Referencing the parameter in a nested UDT will return the name of the nested UDT. In 7.9, the parameter would always return the name of the root UDT.
7. Check your project for script-based tag editing
  - 7.1 The scripting functions for editing tags (system.tag.editTag , system.tag.addTag, etc.) have been deprecated and replaced by system.tag.configure
    - 7.1.1 Follow Ignition's documentation for the new function to change out old functions
8. Check your project for system.alarm.listPipelines() function
  - 8.1 This 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", but it now accepts a project name as a parameter, allowing the function to check for pipelines in any project.
9. Ignition version 8.1.6 started embedding Java version 11.0.11. This could impact the

following since the new java version drops support for TLS v1.0 and v1.1:

- 9.1 Database connections:
  - 9.1.1 If the database connection is failing, check your database vendors JDBC driver recommendations. Updating the driver may resolve the issue.
- 9.2 HTTP calls from Ignition
  - 9.2.1 This is unlikely since modern browsers and services already deprecated older versions of TLS, but if this is an issue then you'd need to update the service (if the endpoint is an internally hosted service) or check the documentation for the service.
10. Check project properties for default tag provider and database and set appropriately if the automatic process misses them
11. If you are importing **SRContext** from **com.inductiveautomation.ignition.gateway** in Ignition version 7.9, note that it has been renamed to **IgnitionGateway** in version 8.X
12. Check your OPC tags to ensure they were updated correctly. If tags are using the "Ignition OPC-UA Server" (with Dash) in newer versions of Ignition, then they have removed the Dash in the default server name.
13. Go through the application and logs and check functionality for anything that was missed

## Deploy the Upgraded System to Production

1. Take backups of the production 7.9.x system
  - 1.1 This includes any databases and Gateway backup
2. Go through and take screenshots or copy the config into notepad of the Gateway settings, database/OPC connections, etc.
  - 2.1 This is to keep track of the settings for various connections and to keep track of what's connected or which devices are enabled on the production server for reference after Gateway restore from the DEV environment
3. Take backups of the Ignition Gateway from your DEV environment and move them to the production environment to prepare for restore later
4. Make note of the Ignition License key
  - 4.1 We will be reusing it later (if customer doesn't have a new key)

5. Deactivate the license key of the production Gateway to allow reactivation later
6. Check if Ignition service has a particular login account set by going into the properties of the services and selecting the “Log On” tab
  - 6.1 If it does, make sure you have the credentials since it will need to be setup again after installation
    - 6.1.1 If you are going to do an upgrade and not a clean install of Ignition, the account credentials should remain.
7. Stop Ignition service
8. Uninstall Ignition (*if upgrading skip this step*)
9. Install newer version of Ignition
  - 9.1 If upgrading, it will ask to proceed with upgrade. Select “Yes”.
    - 9.1.1 It will attempt to upgrade the installed modules, so have the newer versions available to point to during the upgrade process
  - 9.2 If you are not upgrading, then proceed with a normal install of Ignition
    - 9.2.1 Install needed Ignition modules (Sepasoft, etc.)
  - 9.3 Add account credentials for the Ignition service if applicable from Step 6
10. Restore the DEV environment Ignition Gateway backup
11. Go through and start re-enabling the connections that you made notes of in Step 2 and point them to the production server
  - 11.1 For the database connected, they may fail due to the newer version of Ignition needing an updated JDBC driver for the version of the database that it is trying to connect to.
    - 11.1.1 Install the appropriate JDBC driver and copy the auth .dll file to the following new location in 8.1:  
**\\Inductive Automation\\Ignition\\lib\\core\\gateway**
    - 11.1.2 You will have to restart the Gateway service for these changes to take effect
  - 11.2 After the database connections are working, and if you are using the Sepasoft modules, you will need to restart the production module for the changes to take effect
12. If there are any OPC’s connected like Kepware that are using SSL or any other security protocol, you will have to re-trust the connects

13. If the client is using LDAPS protocol for A=active directory in Ignition 8.1, the CA certificate location has changed
  - 13.1 Add the certificate to this new directory in the Ignition installation folder. The location is below:  
**data/certificates/supplemental**
14. Go through the application and logs and check functionality for anything that was missed
15. The Vision Client launcher has changed in 8.1.x, so be sure to plan accordingly to update the clients and Kiosks to the new version