Installation & Upgrade Guide

Kryon Process Discovery

V. 21.6

Document revision: 16-Jul-2021

Contents

APPENDIX C: Technical Data

Introduction

Purpose

This document is the User Manual of the Process Discovery Console. It is intended to provide all the necessary information to use this software to discover processes in your organization.

Scope

This guide covers the Process Discovery console configuration end-to-end.

Intended audience

This guide is intended for business analysts in-charge of process discovery and process mapping.

In this Chapter:

What is Kryon Process Discovery™?

Kryon Process Discovery is an automated, data-driven AI technology which finds, maps and documents existing businesses tasks in near real-time. The **Discovery Robot** working silently on employee's computers records every activity the employee does throughout his workday.

By examining the digital footprint left by users, a detailed representation of the business process is automatically created. The data is then analyzed to recommend and create process models, called wizards.

Business analysts can then use the **Kryon Process Discovery Console** to develop these wizards and export them to **Kryon RPA Studio** to be fine-tuned and tested by RPA developers and automated.

System Architecture Components

Discovery Robots

Lightweight clients installed on employee desktops that silently monitor business-related activities without impacting end-user productivity. They provide full visibility into all business activities at the application level by collecting behavioral data about every user, process, and application across the entire business unit or organization – even when the user's computer is off-network and offline. The data collected by **Discovery Robots** (a screen shot/metadata for each user action) is sent to the **Discovery Server** for analysis. The raw data collected by the **Discovery Robot** is comprised of:

1. A screenshot for each user action; *and*
2. Detailed metadata corresponding to each screenshot, including –
    - Application name
    - User name
    - Event type (e.g., mouse wheel, left mouse click)
    - Mouse position (e.g., x:933, y:637)
    - Time stamp

The System Admin defines the desktop and web applications that are monitored by the **Discovery Robots**.

Discovery Server

The **Discovery Server** utilizes the data collected by the robots perform complex algorithmic processes, including:

- **Image Analysis** – extraction of relevant information from every screen shot
- **Image Clustering** – identification of repeated actions
- **Discovery** – Identifying highly repeated processes and calculating statistical information on duration, actions, applications, etc.
- Output of process and variant data to the **Process Library**

The **Discovery Server** includes **Application Databases**. These are the databases (either MariaDB or MySQL) in which all the data collected by the **Discovery Robots** is stored. The data collected by the **Discovery Robots** is immediately encrypted and transferred to the application databases and remains on the client machine for a short time.

Process Discovery User Management Tool

Kryon Process Discovery User Management Tool grants user access to the Process Library (**Keycloack Service**).

Process Discovery Console

A browser-based application providing an overview of discovered processes selected and saved by the Business-Analyst (you), with the ability to drill down into all the underlying details.

Using the Process Discovery Console you can:

1. Set and configure the collected data:
    - Manage Teams and user access
    - Define applications for discovery
    - Manage the recorded data
    - Manage the **Discovery Robots** and their licenses
2. Discover the best candidate processes for automation
3. Select and add the desired processes to the **Process Library** for further analysis and for mapping
4. Generate processes files for automation (used in **Kryon Studio**) and supporting documents.

The Process Discovery Console can be accessed using the Chrome or Edge web browsers from any machine with access to the **Discovery Server**.

About Automation and Integration

Integration with RPA studio

RPA Studio is an Integrated Development Environment (IDE) that enables easy creation and editing of simple and advanced automation wizards.

The integration between the **Process Library** and **Studio** allows managers to send processes directly to automation as pre-developed wizards, including wizards steps, action variations, decision points, and application data manipulations. Automation developers can then use Studio's intuitive interface and robust toolbox of available commands to make any necessary revisions.

Installation Overview

The major steps in installing Kryon Process Discovery are:

1. Installing the Kryon Process Discovery Server
2. Installing Discovery Robots on employee workstations
3. The default communication protocol is HTTP. Want to configure HTTPS communication? See Process Discovery over HTTPS

Installation Package Contents

1. **Kryon Process Discovery Server installation files**
   - `PDServer64BitSetup.exe.json` - Kryon Process Discovery Server Installation configuration file. See Discovery Server Installation Configuration File
   - `PDServer64BitSetup.exe` - Kryon Process Discovery Server installation executable file
   - `[License file name].llk` - The license file. it may come with the installation package; in which case you can install it when you install the server. Otherwise you can install it later, when it is available
2. **Discovery Robot Installation File**
   - `DiscoveryRobotSetup.exe`
3. **Kryon Process Discovery Documentation**
   - Kryon Installation Guide (this manual)
   - Kryon Process Discovery User Guide
   - Security Overview
   - System Architecture
   - Release Notes

Both **Kryon Process Discovery Server** and **Discovery Robot** Installation packages install additional Windows components and software to the server machine. See Installed Components & Software

System Requirements for Discovery Server

Hardware & software requirements

Item	Requirement
Processor (minimum)	Intel i7 or Xeon / 16 core* Processors with AVX support
RAM (minimum)	32GB
Free disk space (**SSD**)	Minimum **150**GB Recommended **500**GB (Local hard disk *only*)
OS	Windows Server 2016 or higher
Database	Server installation installs MongoDB 4.4.3 and MariaDB 10.4.7. Alternatively, you can connect to a preexisting dedicated **Windows version** database server (MariaDB 10.3.7 or higher / MySQL 8.0.11 or higher / MongoDB 4.4.3 or higher)
Supported browser for accessing Process Library	Chrome v69 or higher, Edge v69
Network bandwidth	500 MB/day per active Discovery Robot client ~15 KB/s per active Discovery Robot client ~1.2 MB average packet size (10 user actions per packet)

*When a single server with 16 cores is not possible, installation on 2 servers with 8 cores each is an option. Contact Kryon Support for details if this is option is required.

TIP
How many cores?

To verify the number of processor cores are installed on a machine:

1. Open the **Windows Task Manager > Performance tab**
2. The **Logical processors** field provides the information you're looking for

Yes, it might seem counter intuitive, but for purposes of **Process Discovery**, it's the **Logical processors** field you're interested in – not the **Cores** field!

System Requirements for Discovery Robots

Item	Requirement
Processor	i3 / 2 cores minimum i5 / 4 cores recommended
Supported workstations	1. Windows desktops 2. Remotely managed workstations, like: • Remote client machines of a terminal server running Microsoft Terminal 2016 and up • Cloud desktops running Amazon WorkSpaces client application
OS	Windows 10 (64- or 32-bit); or Windows 7 (64- or 32-bit)
Supported browsers for Discovery Robot recording	• Chrome v69 or higher • Edge v17 or higher • Internet Explorer v11 or higher • Mozilla Firefox 63
Supported languages for multi-language keyboard recording	*Albanian, Latvian, Armenian, Lithuanian, Bulgarian, Macedonian, Catalan, Norwegian, Croatian, English, Polish, Czech, Portuguese, Danish, Romanian, Dutch, Russian, Estonian, Slovak, Finnish, Slovene, French, , Spanish, German, Swedish, Greek (Modern), Turkish, Hungarian, Ukrainian, Icelandic, Russian, Italian, Hebrew*
Network bandwidth	500 MB/day per active Discovery Robot client ~15 KB/s per active Discovery Robot client ~120 KB average packet size (1 user action per packet)

Installing the Kryon Process Discovery Server

The following sections describe steps to install **Discovery Server** and databases on a single server. For guidelines for a silent installation, refer to Silent Installation of Discovery Server .

In this Chapter:

Discovery Server Installation Steps

Follow these steps to install the **Discovery Server**:

Step 1: Verify Server Requirements

Ensure that the Server meets the required hardware and software specifications.

Step 2: Open Network Ports

1. The following **default external ports** are opened automatically in the Windows Firewall during the **Discovery Server** installation process. These ports are intended for an installation that *does not use* SSL/TLS (see below for SSL/TLS ports). Open them in your hardware firewall prior to installing the server

Default Port (Not using TLS)	Port Used by	Port Used for
80	NGINX	Communication between the **Discovery Robots** and Console users with the **Discovery Server**

If you *will* be using SSL/TLS , open the following ports in your hardware firewall prior to installing the server.

NOTE: Only SSL/TLS v1.2 is supported.

Default Port	Port Used by	Port Used for
443	NGINX	Communication between the **Discovery Robots** and Console users with the **Discovery Server**

2. (optional) You can override the default port setting if you want. At this time, make sure the ports you choose to use are open in the hardware firewall. Be sure to write down the port numbers you use; later you will need to configure them in the installation settings.
3. (optional) The following *default backend ports* are used in the **Discovery Server**.We recommend *NOT* to open them in the firewall. They are customizable, if required:

Default Port	Port Used for
5058	Communication between the **Discovery Server**and the **User Management Tool**

Default Port	Port Used for
50100-50130	Communication between the **Discovery Server** and the **Discovery Console**
51000-51030	Communication between the **Discovery Server** and the **Discovery Console**

Step 3: Kryon Process Discovery Database Engine

The **Discovery Server** installation package automatically installs MariaDB 10.4.7 and MongoDB 4.4.3.

You can opt to connect to a preexisting database engines.

Databases are installed on the **Discovery Server** to collect, analyze, and work with the data received from the robots. Kryon Process Discovery uses four application databases:

Database	Database engine	Created users
PD application database #1 (data segregated by Team)	MariaDB 10.3.7 (or higher) or MySQL 8.0.11 (or higher)	The Process Discovery server installation package automatically installs MariaDB which creates a database called **kryon-db** and the following default users: • **root** • **pdbdev**
PD application database #2 (data aggregated for all Teams)		
Process Library database		
User Management platform database		
Raw data collections: Images + metadata recorded by Discovery Robots (segregated by team) collection prefix: `pd-rawdata-${TEAM_GUID}`	MongoDB 4.4.3 (or higher)	The Process Discovery server installation package automatically installs MongoDB which creates a database called **kryon-db** and the following default users: • **kryon-admin** ('root' role) • **kryon-rw** ('readWrite' role)
Masked Images:		

Database	Database engine	Created users
Only if masking is enables (segregated by team) collection prefix: `pd-masking-rawdata-${TEAM_GUID}`		

- If you want to manually install and/or configure MongoDB, Robo 3T, MySQL or MariaDB, see Manually Installing the Database Engines

Step 4: (optional) Update RabbitMQ Permissions

The **Discovery Server** installation package automatically installs RabbitMQ Server.

You can opt to connect to a preexisting RabbitMQ installation, if you have one, later in this installation procedure.

OPTIONAL STEPS:

- If you want to connect to a preexisting RabbitMQ installation from a **Discovery Server** installation, you may need to add permissions to the admin RabbitMQ user at this time. To do so, follow the steps in Add permissions to admin RabbitMQ user for preexisting RabbitMQ version.

Step 5: Copy Installation Files to Local Folder

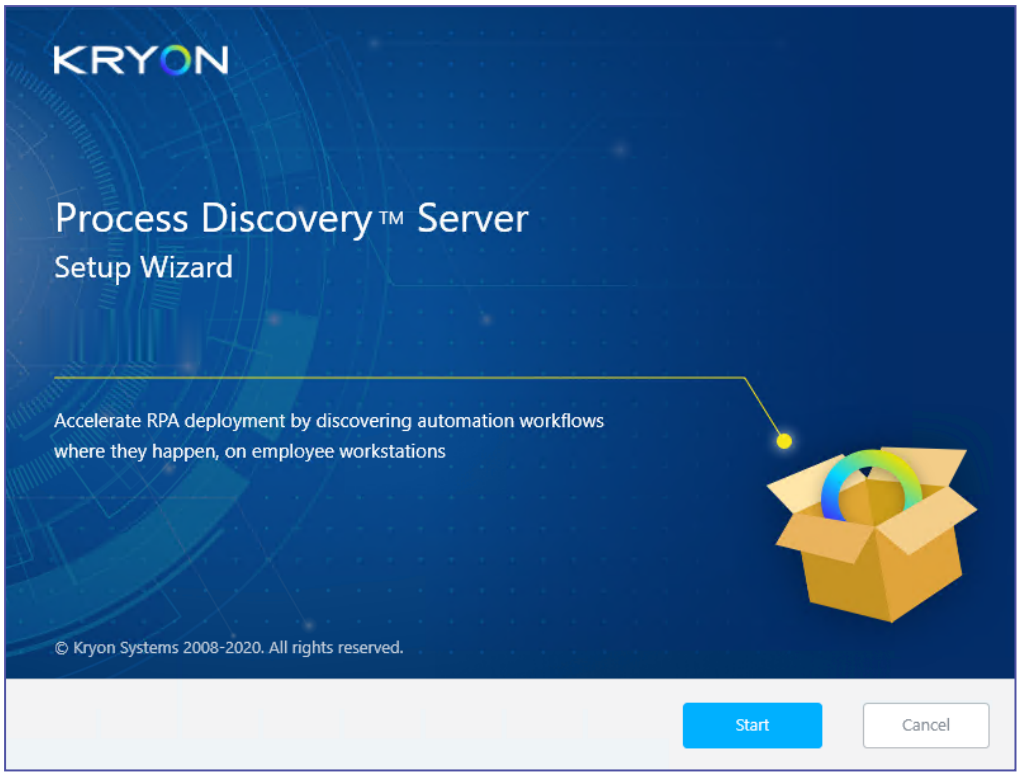
1. Download the Kryon Process Discovery Server installation file, `PDServer64BitSetup.exe`, to a local folder on the Server.
2. If you have received your Kryon Process Discovery license file from your contact at Kryon or the Kryon distribution partner, download it also.

If you don't have a license now, you can install it **after Discovery Server** installation is completed. See Step 10: (optional) Install Kryon Process Discovery License After Installation.

Step 6: Run Kryon Process Discovery Server Setup Wizard

1. Go to the folder where you copied the installation package on the local drive and double click the installation executable file, `PDServer64BitSetup.exe`, to open the **Kryon Process Discovery Server Setup Wizard**.

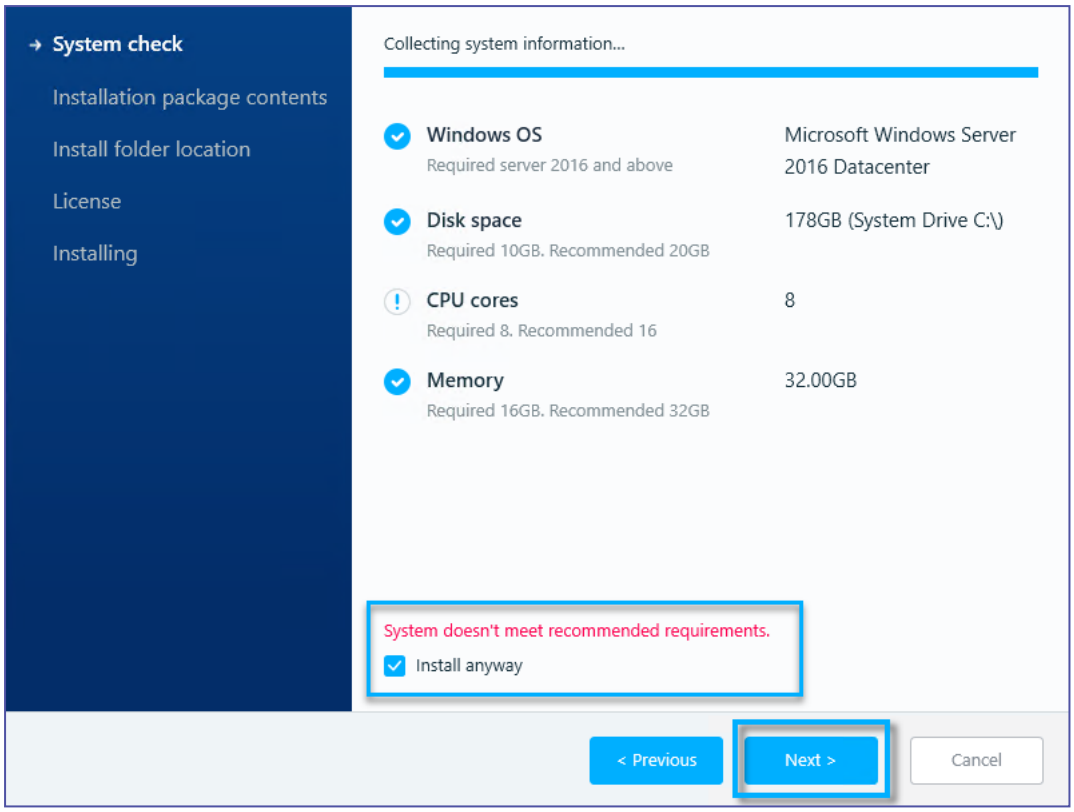
2. Click **Start**. The Setup Wizard begins.



System check

The **Setup Wizard** collects system information from the server to verify compliance with system requirements.

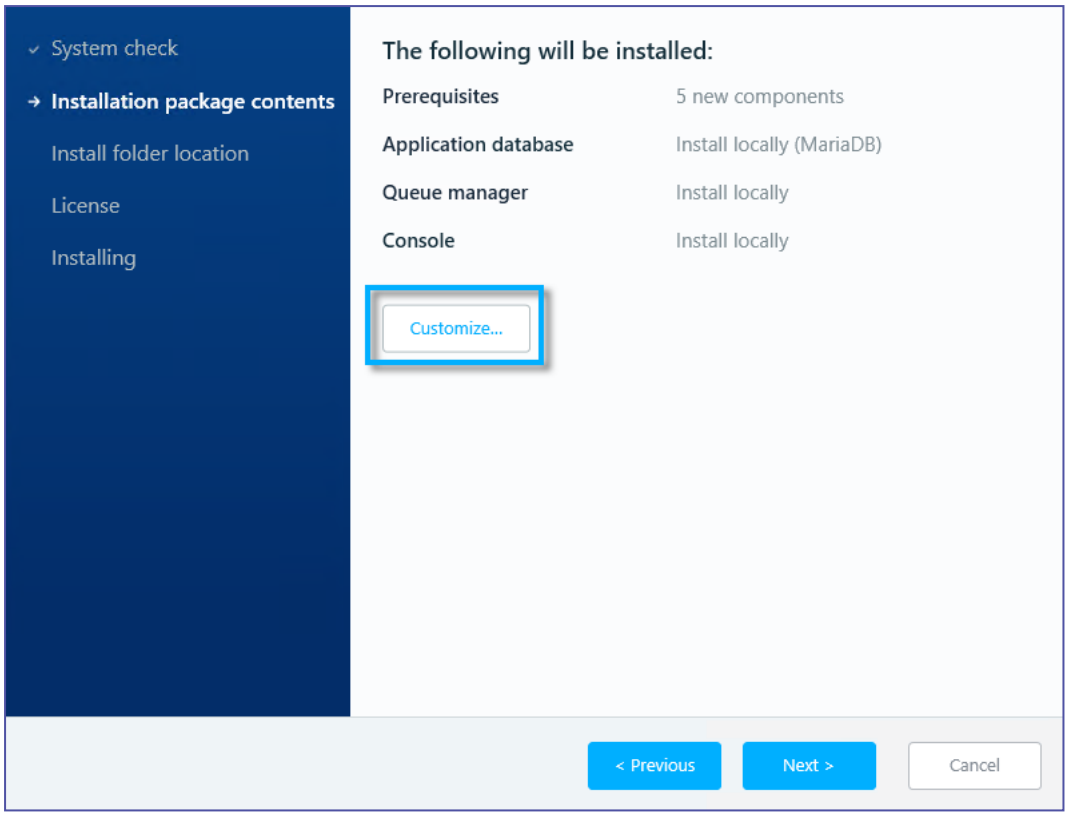
If the system does not meet minimum requirements, the wizard automatically cancels installation



- If your machine doesn't meet recommended system requirements, but *does meet minimum requirements*, select **Install anyways** to continue the **Setup Wizard**
- Click **Next**

Step 7: Installation Package Contents

Default installation package contents are displayed.

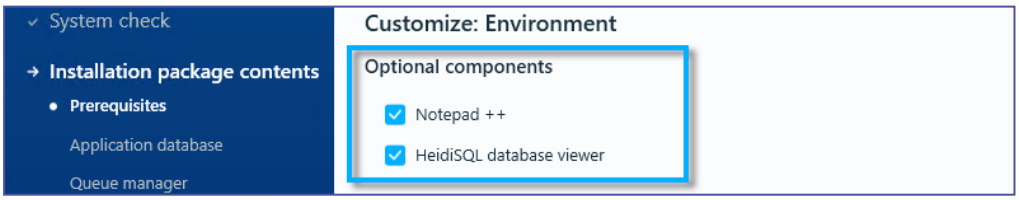


- **TO ACCEPT THE DEFAULT VALUES, CLICK NEXT AND SKIP TO STEP 8: SET INSTALL FOLDER LOCATION; *OTHERWISE*, CLICK CUSTOMIZE AND FOLLOW THE STEPS BELOW:**

Customize prerequisites

Mandatory components cannot be customized, however, you will probably want to install Notepad++ and the HeidiSQL Database viewer, if not already installed.

1. Select the optional components you want to install



2. Click **Next**

Customize application database

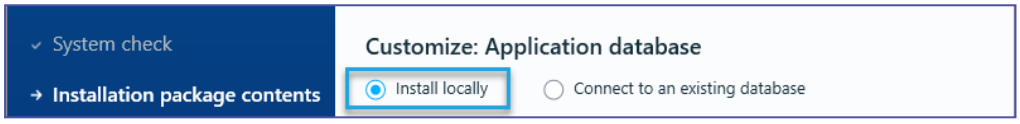
If you want to configure the MariaDB database engine that's automatically installed as part of the installation package, you can use this window to:

- Override the default port `3306`
- Provide strong passwords for the two users created: root and pdbdev (the default for both is `Kryon2020!`)

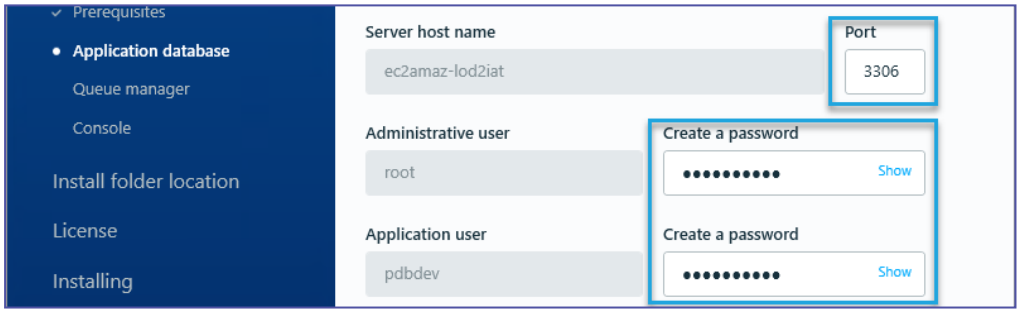
Or, you can use this window to connect to a pre-installed database engine.

To configure the automatically installed the application database engine:

1. Select **Install locally**



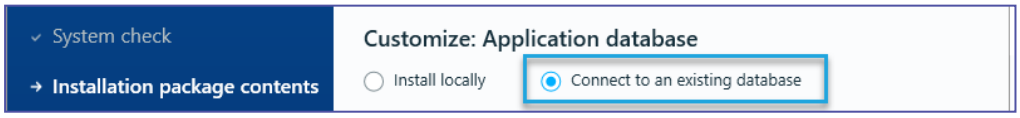
2. Leave the default port `3306` unchanged or enter another port number
3. Provide strong passwords for both the `Root` and `Pdbdev` usernames



4. Click **Next**

To connect to a pre-installed application database:

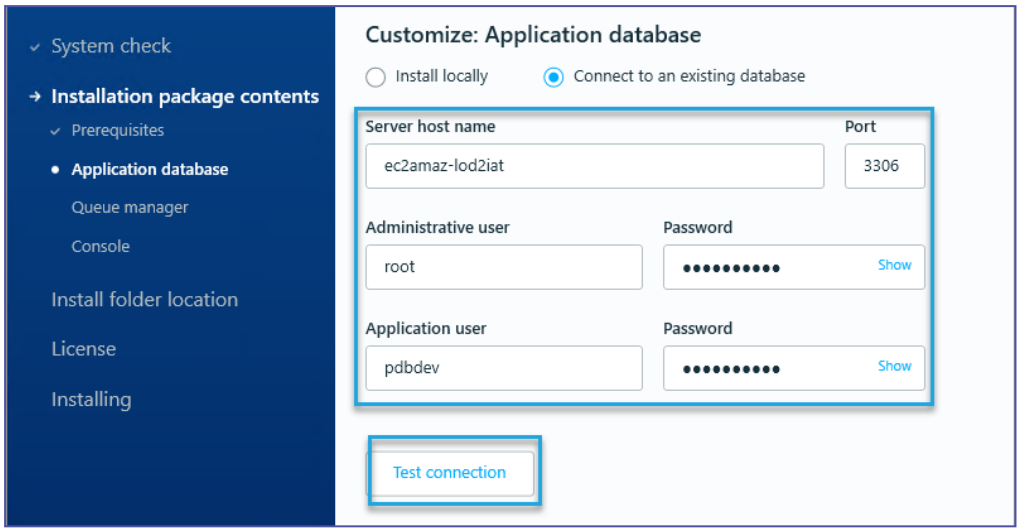
1. Select **Connect to an existing database**



2. Enter the pre-installed Server Host Name
3. Enter the port number of the existing database
4. Enter the user names and passwords for both the Administrative and applications users

   Note: The Administrative user needs full permissions to create schema and

tables.



5. Click **Test Connection** to verify that your server connection is working
6. Click **Next**

Customize queue manager

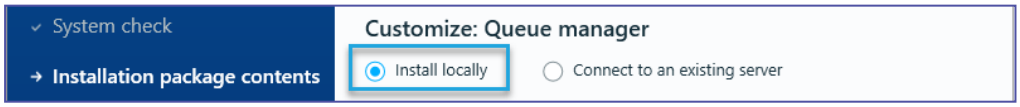
If you want to configure the queue manager (RabbitMQ server) that's automatically installed as part of the installation package, you can use this window to:

- Override the default port `5672`.
- Provide a strong password for the admin user (the default is `Kryon2020!`)

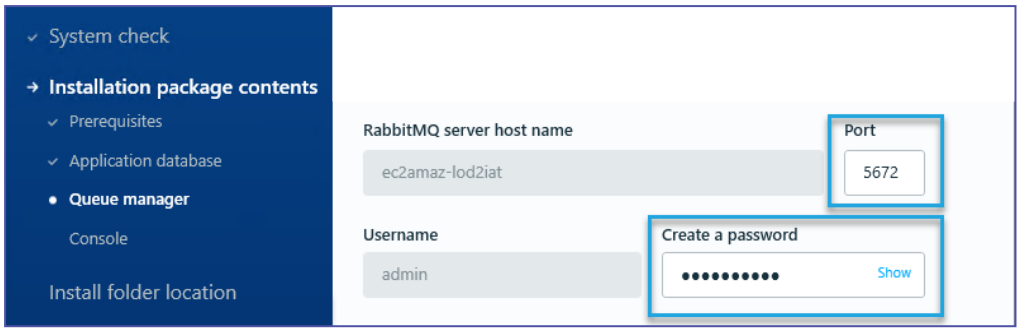
Or, you can use this window to connect to a preexisting RabbitMQ installation

To automatically install the queue manager:

1. On the Customize Queue Manager window, select **Install locally**



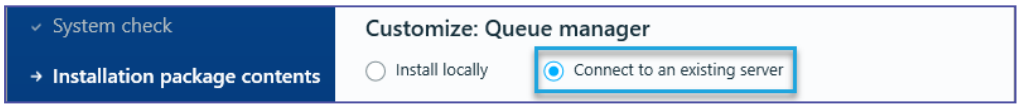
2. Leave the default port `5672` unchanged or type in your own port number (for SSL/TLS change port to `5671`)
3. Provide a strong **password** for the admin user name



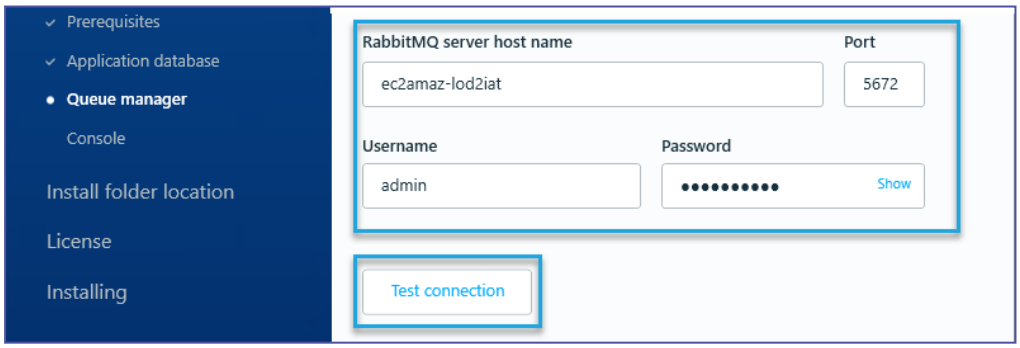
4. Click **Next**

To connect to a preexisting queue manager instance:

1. On the Customize Queue Manager window, select **Connect to an existing database**



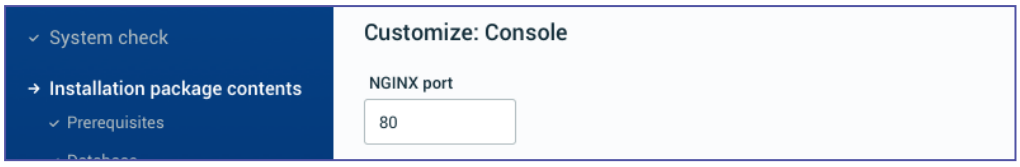
2. Type in the **port** number of the preexisting RabbitMQ server
3. Enter the **RabbitMQ Server Host Name**
4. Enter the **admin username** and provide a strong **password**
5. Click **Test Connection** to verify that your server connection is working



6. Click **Next**

Customize Console

1. Leave the default port `80` unchanged or type in your own port number (to work with SSL/TLS use port `443`)

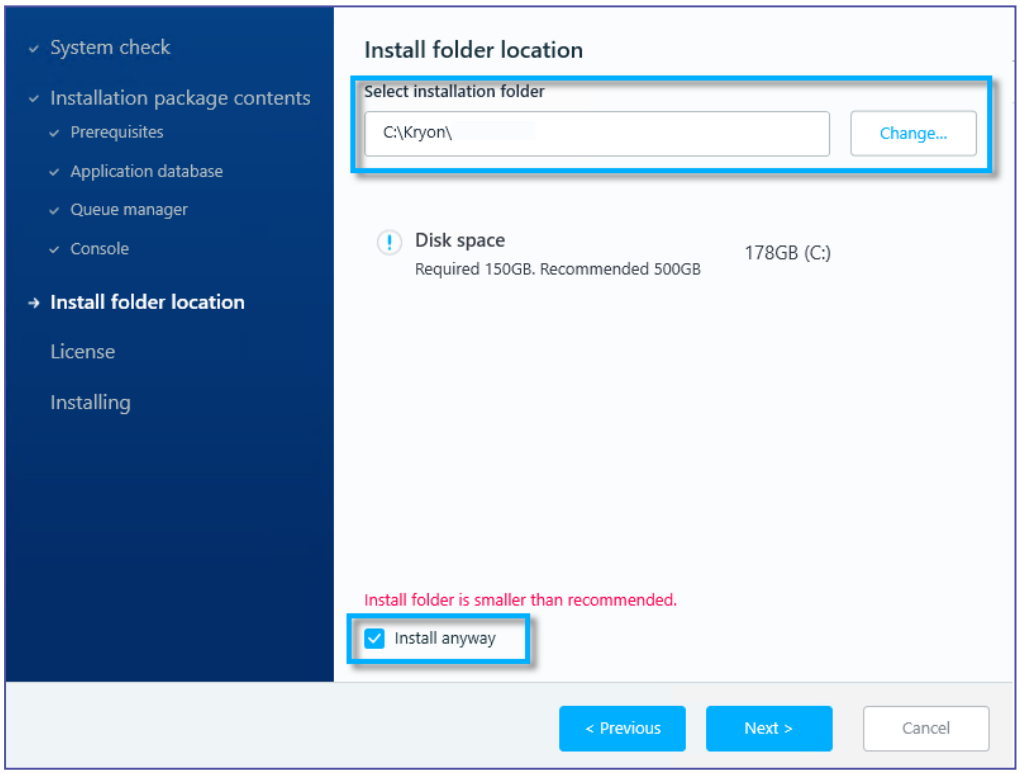


Click **Next**

Step 8: Set Install Folder Location

By default the InstallFolder location is `{local fixed drive with the most free space}:\Kryon\`.

1. If you want to select a different Install folder location than the default, Click **Change** and select the new location.
2. If the selected folder doesn't meet recommended disk space, but does meet minimum requirements, you can continue the **Setup Wizard** by selecting **Install anyways**



3. Click **Next**

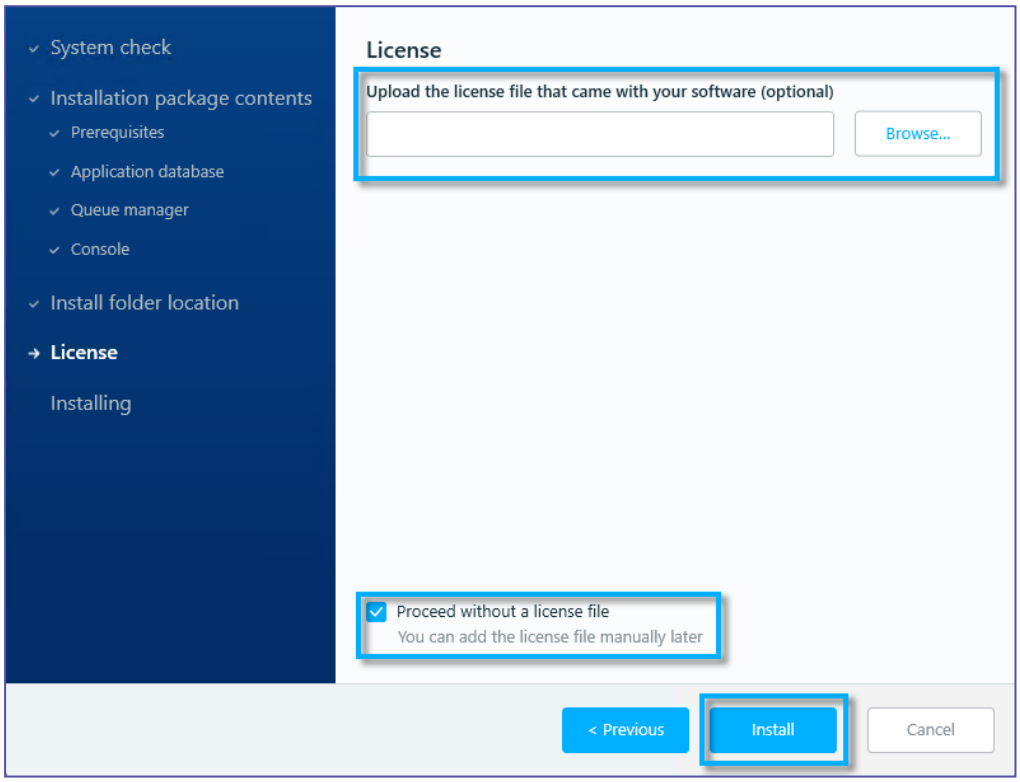
Step 9: Add License

If you have received your Kryon Process Discovery license file from your contact at Kryon or the Kryon distribution partner, and you downloaded it to the server in Step 5: Copy Installation Files to Local Folder, you can install it now.

If you don't have your license yet, no worries! You can follow the steps in Step 10: (optional) Install Kryon Process Discovery License After Installation, when you have your license.

1. If you have already downloaded a license file you can upload it now. Click **Browse** and select the file; *or*

   Select **Proceed without a license file**.



2. Click **Install**. During installation, the wizard provides status updates so that you can monitor the installation progress as setup continues.

   When the wizard finishes, the Discovery Server is installed, but there are still a few more steps you need to do on the Server before you're done:

Next Steps

- Step 10: (optional) Install Kryon Process Discovery License After Installation
- Step 11: Configuring TLS Discovery Server
- Step 12: Verify Installation

Step 10: (optional) Install Kryon Process Discovery License After Installation

Follow these steps to set up your **Process Discovery** license (if you did not do so previously during installation):

1. Copy the license file you received to the Install Folder you set in step 7
2. From the **Windows Services** app or the **Windows Task Manager > Services** tab:
   - Restart the **Kryon Server - Process Discovery Service**
   - Check the **Status** column of the **Kryon Server - Process Discovery Service** to ensure that it shows `Running`

NOTE

If you have not yet received your license file, check with your contact at Kryon or the Kryon distribution partner with whom your organization is working.

Step 11: Configuring TLS Discovery Server

(optional) If you will be using SSL/TLS to secure communication between the **Discovery Robots** and the **Discovery Server**, you should configure it now.

NOTE: Only SSL/TLS v1.2 is supported.

OPTIONAL STEPS:

- Configuring SSL/TLS on the Server

Step 12: Verify Installation

Verify the **Discovery Server** installation by checking that you can connect to the following components:

NOTE

If you're connecting remotely to the **Discovery Server** or if you defined the server's FQDN when you installed it, enter the FQDN of the server instead of `localhost` in the URLs in the following steps:

1. **Connect to RabbitMQ Management Plugin**

   From a web browser on the Discovery Server, enter the following URL:

`http://{FQDN of Server}:15672`

Log in to RabbbitMQ using the username `admin` and the default password `Kryon2020!`

2. **Connect to Process Discovery Console:**
   a. From a web browser on the Discovery Server, enter the following URL: `localhost/console`
   b. Login with the following username and password:

      Username: `Pdconsole`

      Default temporary password: `Pd123456!` (if you changed the password in step 4, use the new password you created)

3. **Connect to SEQ:**

   From a web browser on the Discovery Server, enter the following URL:

   `http://localhost/seq`

4. **Connect to Process Discovery User Management Tool:**
   a. Open an incognito window in Chrome

      Enter the following URL: `localhost/auth/admin/kryon/console/#/realms/kryon`
   b. Login with the following username and password:

      Username: `authadmin`

      Default temporary password: `Kryon123!` (you may be asked to change the password. If you do, be sure to write it down)

If you have connected successfully to the above, you're done installing the server!

Next Steps

Install the Discovery Robots

Troubleshooting Discovery Server Installation - Logs

The **Discovery Server** installer records detailed logs of the entire installation process, including all components. These logs can be a useful resource for troubleshooting.

TO LOCATE THE LOGS:

1. Right-click the **Windows Start** button
2. Select **Run**
3. Type `%temp%`, then hit <ENTER>
4. A Windows Explorer window opens to the logged-in user's `%temp%\` folder
   - If you are logged into a local machine, the logs are in this folder
   - If you connected remotely, the logs are located one directory level up

The bundle setup log is in the relevant Temp folder and its name is `PDServer64BitSetup_{execution timestamp}.log`.

Installing Discovery Robots

Discovery Robots run silently in your employees' machines, collecting data on how they utilize business applications to perform their daily tasks.

The following sections describe steps to install **Discovery Robots** in employee workstations. There are two option for installing **Discovery Robots** on the employee machines:

- Silent Installation of Discovery Robots ; *or,*
- UI Setup Wizard of Discovery Robots

In this Chapter:

Before Installing a Discovery Robot

1. Before you start installing Discovery Robots, your organization should provide you with a list of client workstation(s) where (missing or bad snippet) will be installed .
2. Ensure that each client workstation meets the required hardware and software specifications
3. Add the pddr.exe to the allow-list of the organizational antivirus software
4. Have available the **Discovery Robots** installation package file – `DiscoveryRobotSetup.exe`
5. Have available the **PDDR.keys** file from the **Discovery Server**

   The keys file is generated as part of the **Discovery Server** installation process. You can find it on the **Discovery Server** in `{InstallFolder}\PDServer\Support`
6. Make a note of the Fully Qualified Domain Name (FQDN) of the **Discovery Server**

TIP

How to find out your server's FQDN

a. Click the Windows **Start** button and enter `cmd` into the text box. Hit <Enter> to open your Windows command line utility

b. Type `ipconfig` and hit <Enter>. This displays the IP address for your Windows server. Use this IP address to view the fully qualified domain name of the server

c. Type `ping -a IP` where IP is the IP address for the computer. Hit <Enter>. The "-a" switch returns the domain name of your server

7. Lastly, you should receive from your organization the specific configuration requirements for each **Discovery Robot**. You will need these requirements handy when you configure the robots after they are installed.

Silent Installation of Discovery Robots

This section describes procedures for a silent installation of the **Discovery Robots**. See UI Setup Wizard of Discovery Robots to install using the UI Setup Wizard.

When you run a silent install of the **Discovery Robots**, you can simply execute the installation file `DiscoveryRobotSetup.exe`; or you can choose to:

1. Run the installation file with a START /WAIT command that let's you see the exit codes of the installation when it completes; *or*
2. Include installation parameters in the command line; *or*
3. Both

If you choose to add parameters in the command line, Silent mode installation supports the inclusion of the following installation parameters:

Parameter Name	Default Value (if unspecified/left blank)	Description
InstallFolder	`C:\Program Files\PDDR`	Folder in which the **Discovery Robot** files will be installed
AddToStartup	`true`	Determines whether the Discovery Robot will run automatically each time a user of the machine logs in to Windows
messagesBrokerHost	`localhost`	{`Discovery Server IP address or FQDN (Fully Qualified Domain Name)`} of the **Discovery Server** • Must be changed from the default value either as an installation parameter or by changing the value in the Discovery Robot configuration file (after installation)
stealthMode	`false`	Determines whether the

Parameter Name	Default Value (if unspecified/left blank)	Description
		robot's icon will appear in the Windows taskbar: • `false` = icon will appear in the taskbar • `true` = icon will not appear in the taskbar (robot will be invisible to user)
persistRecords	`false`	Determines whether a copy of the robot's will be maintained on the robot (in addition to syncing to the server). Setting this parameter to `true` is recommended for debugging purposes only.
START_ROBOT	`true`	Launches the **Discovery Robot** after installation
CaptureScreenshots	`true`	Determines whether screenshots are collected when recording. Setting this parameter to `false` allows recording in a 'metadata only' mode, which is collecting the data of application usage and work-patterns without capturing screenshots.
PDDR_KEYS_PATH		
standaloneMode		
HashUserName		`true` / `false`
messagesBrokerProtocol		`https` / `https`

Parameter Name	Default Value (if unspecified/left blank)	Description
messagesBrokerPort		80 / 443

TO RUN THE DISCOVERY ROBOT INSTALLATION PACKAGE SILENTLY:

1. Access the command prompt and run **as an administrator**
2. Change directory to the location of where you copied the installation files by typing in:

   `CD {Folder location}`

   For example, `CD C:\FolderName\`
3. Hit <ENTER>
4. Based on your installation preferences:
   a. To simply run the installation file, type in: `DiscoveryRobotSetup.exe -silent`
   b. To see the exit codes of the installation when it completes, type in: `START /WAIT DiscoveryRobotSetup.exe -silent`
   c. To specify parameters to either of option a or b above: add the parameters to the command line. The command line would look like this when parameters are specified:
   - When using START /WAIT:
     `START /WAIT DiscoveryRobotSetup.exe [Parameter1=Value1 Parameter2=Value2 Parameter3=Value3] -silent`
   - When not using START /WAIT:
     `DiscoveryRobotSetup.exe [Parameter1=Value1 Parameter2=Value2 Parameter3=Value3] -silent`

NOTE

Parameters must be specified using exact parameter names. For example, `DiscoveryRobotSetup.exe /Silent InstallFolder="C:\The folder\The Company\The PDDR"`

6. During installation the system may restart one or more times. Following restart, Discovery Robot installation will resume right where it left off.
7. You can view an exit code following installation when you install using a START /WAIT command, as explained above:

- Immediately after installation completes, run the **ECHO** command with the **%errorlevel%** parameter, by typing in the command line: `ECHO %errorlevel%`.

  The exit codes are shown in the command line as follows:

  - 0 = success
  - 1602 = user canceled
  - 1641, 3010 = success, but must reboot to finish install

7. Continue by configuring the Discovery Server address and additional Discovery Robot settings as required.
8. When done configuring **Discovery Robot** settings, restart the robot
   a. Right-click the robot's tray icon, and select **Quit**
   b. Run `{MainRobotFolder}/pddr.exe` to start the robot

NOTE

The {MainRobotFolder} is the folder in which the files were installed:

- By default, this folder is `C:\Program Files\PDDR`
- If you specified a different location during installation, the `{MainRobotFolder}` is the folder you specified

UI Setup Wizard of Discovery Robots

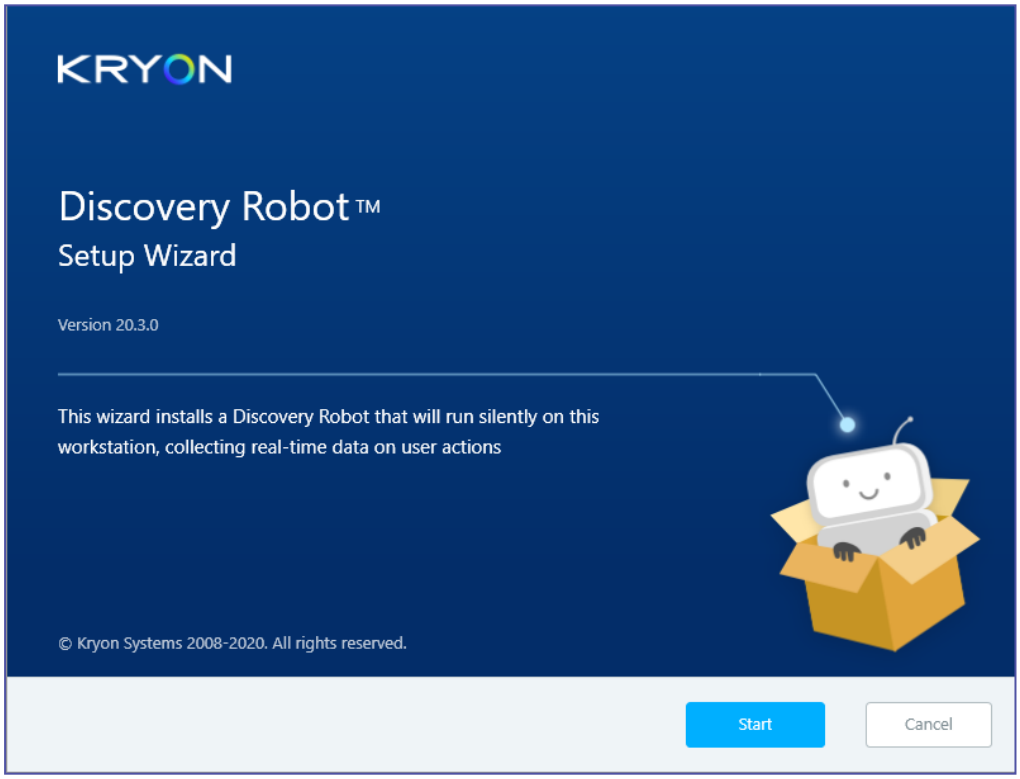
Follow these steps to install a **Discovery Robot** in UI mode. For instructions for a silent install, see Silent Installation of Robots.

TO INSTALL A DISCOVERY ROBOT ON EACH CLIENT MACHINE:

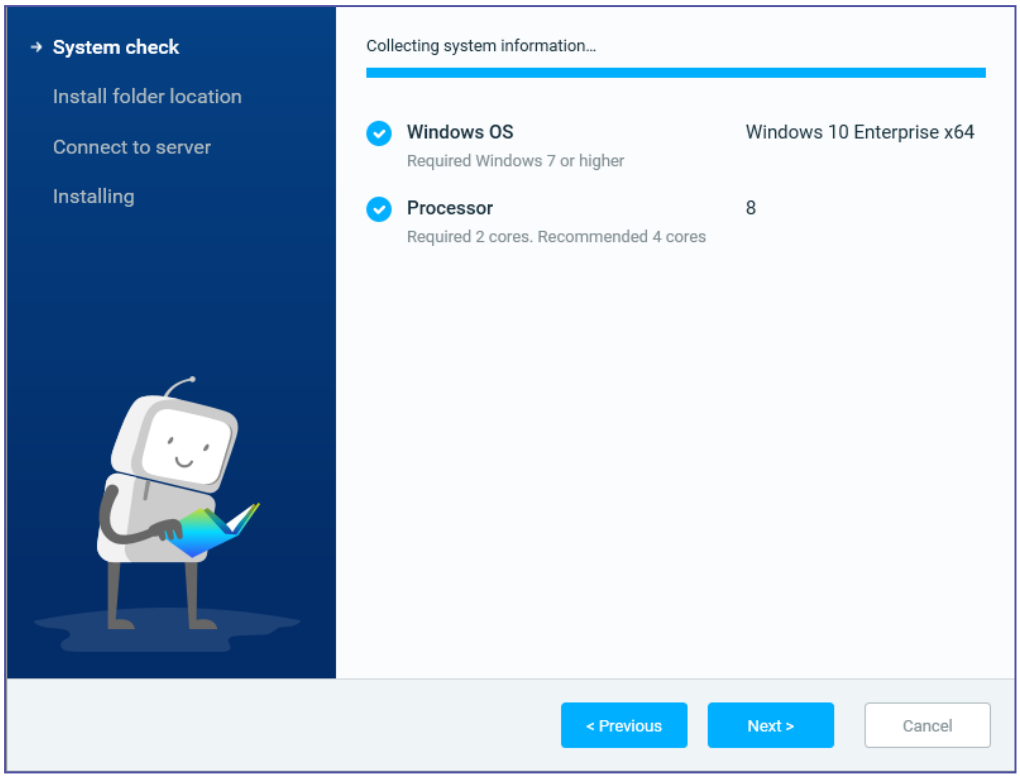
1. Copy the installation file **DiscoveryRobotSetup.exe** and the **PDDR.keys** file to the same folder on the employee machine.

   (The keys file is generated as part of the **Discovery Server** installation process. You can find it on the **Discovery Server** at `{InstallFolder}\PDServer\Support`).

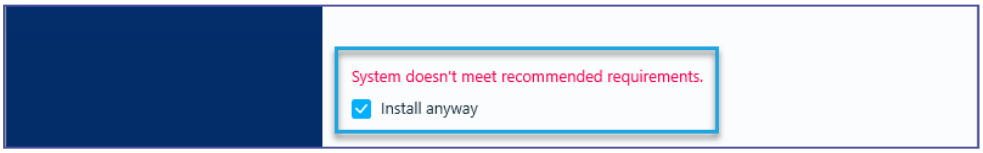
2. Right-click the file `DiscoveryRobotSetup.exe`, and select **Run as administrator**
3. The Kryon **Discovery Robot** Installation welcome screen opens:



4. Click the **Start** button. The System Check screen opens.

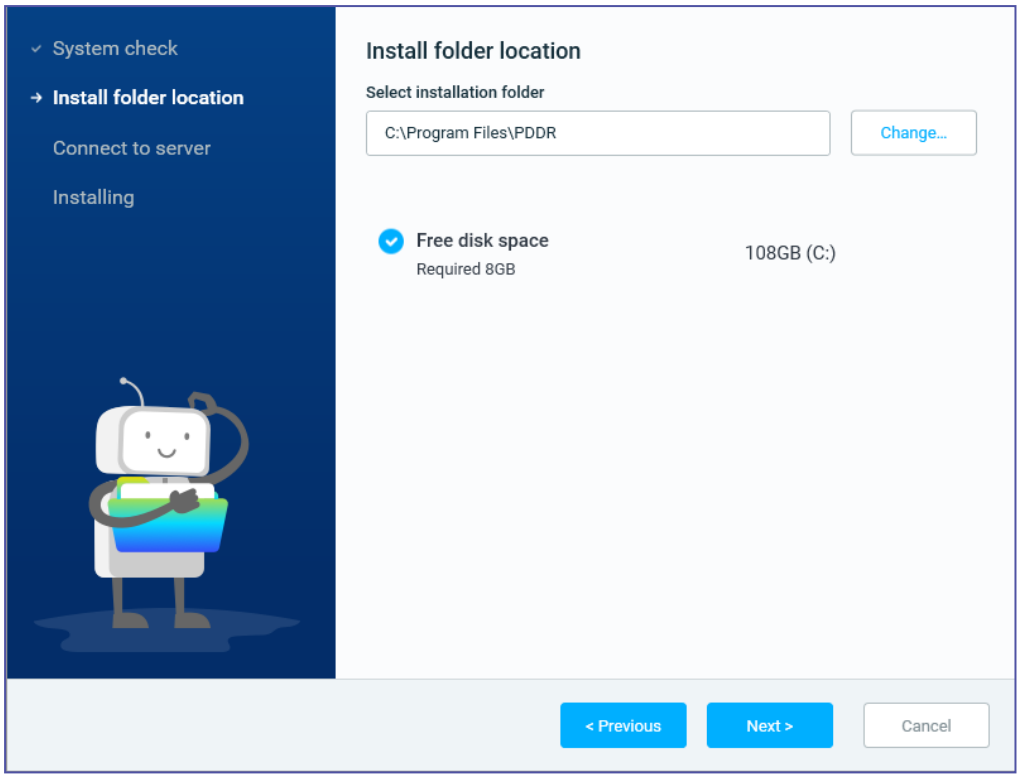


5. The **System Check** collects system information to verify compliance with system requirements.
   a. If the system does not meet minimum requirements, the wizard automatically cancels installation.
   b. If your machine doesn't meet recommended system requirements, but *does meet minimum requirements*, you can continue the **Setup Wizard** by selecting **Install anyways**.

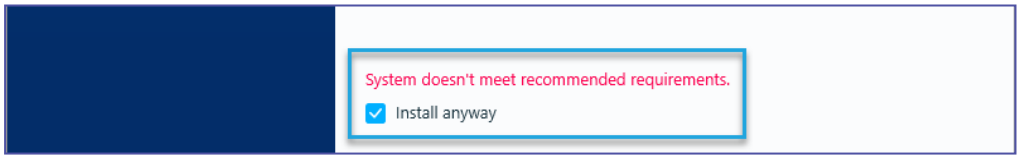


6. Click **Next**. The Installation Folder screen opens.

7. By default, the **Discovery Robot** files are installed on the client workstation in `C:\Program Files\PDDR`. If you want to change the installation folder location, click **Change** and select another location.



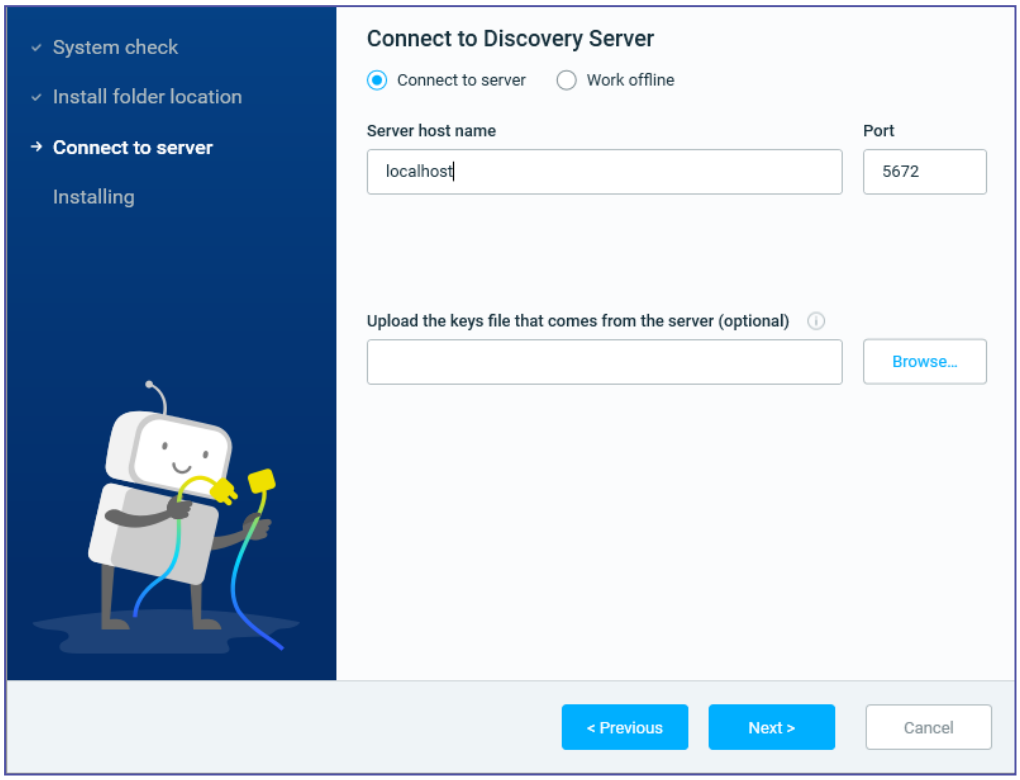
8. If your machine doesn't meet recommended free disk space, but *does meet minimum requirements*, you can continue with the selected install folder location by selecting **Install anyways**



9. Click the **next** button. Installation begins

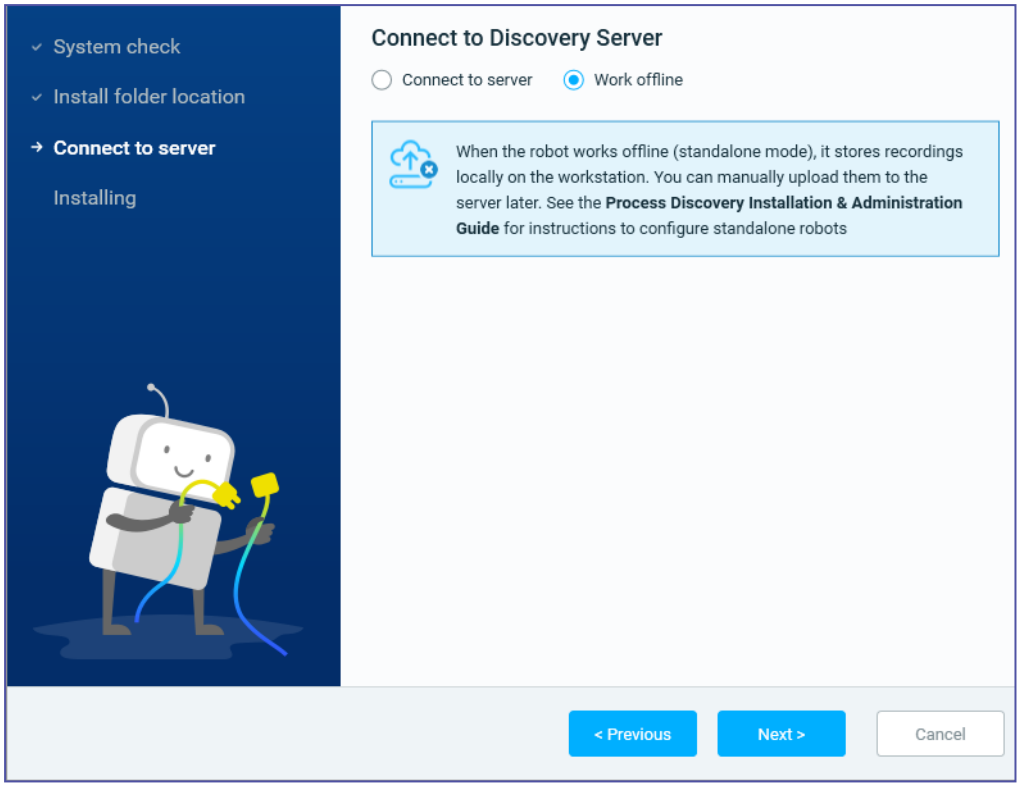
10. Here you have two (2) options, to **Connect to server** or to **Work offline**.

Connect to server: Configure a connection to a dedicated Process Discovery server by entering the server IP address or FQDN (Fully Qualified Domain Name). You can also browse the **PDDR.keys** file and upload it.

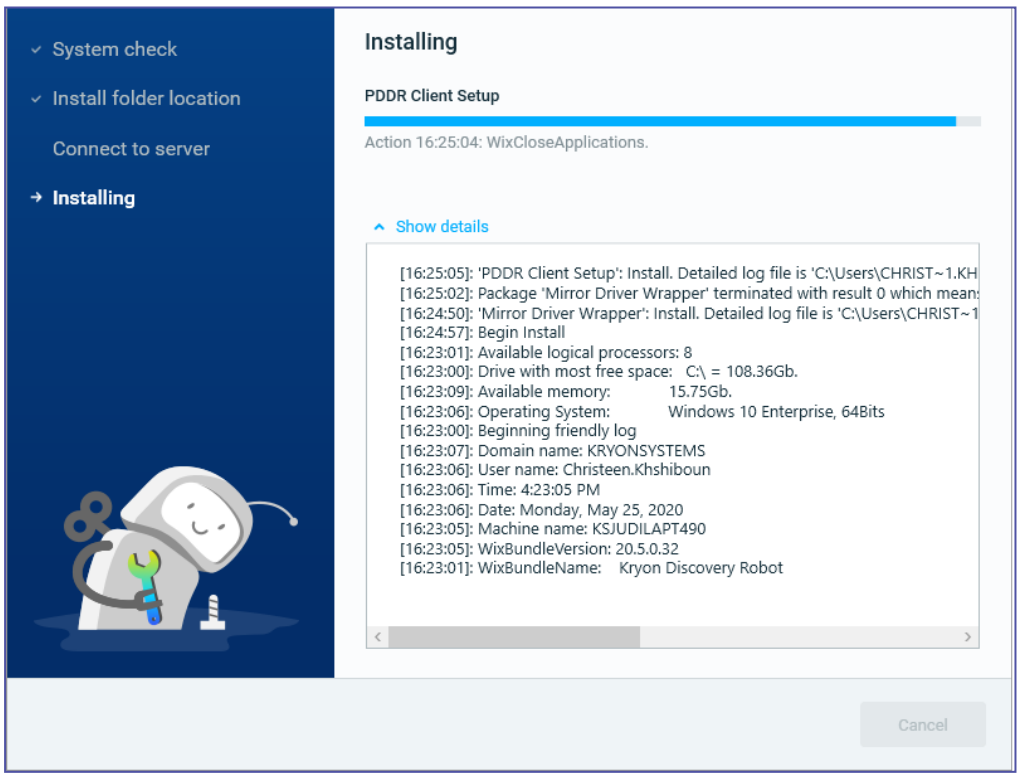


Work offline: Configure the **Discovery Robot** to work in a standalone mode (offline from the server). For additional configuration and administration of

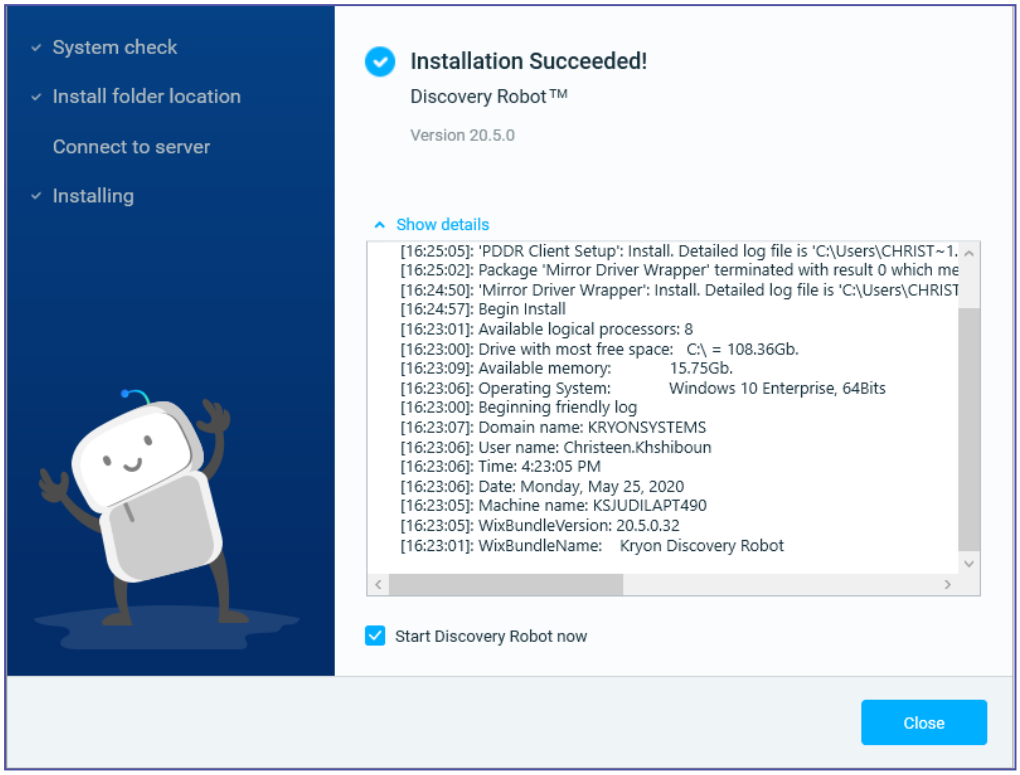
standalone robots, see the ***Standalone Discovery Robots*** in the User Guide.



11. The computer may restart one or more times during installation.
Following restart, **Discovery Robot** installation will resume right where it left off.



12. When installation is complete, the Success message appears:



13. Continue by configuring the Discovery Server address and additional Discovery Robot settings as required.

14. When done configuring **Discovery Robot** settings, restart the robot

    a. Right-click the robot's tray icon, and select **Quit**

    b. Run `{MainRobotFolder}/pddr.exe` to start the robot

> **NOTE**
>
> **The {MainRobotFolder} is the folder in which the files were installed**
>
> - By default, this folder is `C:\Program Files\PDDR`
> - If you specified a different location during installation, the `{MainRobotFolder}` is the folder you specified

Configuring Discovery Robots

The basic Discovery Robot setting are configured upon installation through the installation wizard. Once **Discovery Robots** are installed on the employee workstation, you can apply additional Discovery Robot configurations by editing the Discovery Robot Configuration File. You should receive from your organization the specific configuration requirements for each **Discovery Robot**.

Discovery Robot Configuration File

You can locate and edit the Discovery Robot settings in the Discovery Robot Configuration File.

The Discovery Robot Configuration file name and location depends on the following:

Before the first run of Process Discovery Robot:
`${InstallationFolder}/PDDR/pddr.appsettings_template.config`
(The usual path is: C:\Program Files\PDDR)

After the first run of Process Discovery Robot:
`%localappdata%/Kryon/ActionsRecorder/config/pddr.appsettings.config`

Parameter	Description/change
`<add key="messagesBrokerHost" value=" FQDN /IP Address"/>`	Sets the address of the **Discovery Server** `<add key="messagesBrokerHost" value="{{Discovery Server IP address or FQDN (Fully Qualified Domain Name)}}"/>` • **MUST** be changed from the default value of `localhost`
`<add key="messagesBrokerPort" value="port#"/>`	Sets the port number of the Process Discovery server. You can edit and change the port number. The default port number is `5672`.
`<add key="stealthMode" value="false"/>`	To hide **Discovery Robot's** tray icon on the machine's end-user(s), change the key value to `"true"`.
`<add key="persistRecords" value="false"/>`	By default, recordings by **Discovery Robots** are not maintained on the employee workstation once they have been uploaded to the **Discovery Server**.

Parameter	Description/change
	To maintain the recordings on the workstation after uploading to the **Discovery Server**, for debugging purposes, change the key value to "`true`".
`<add key="HashUserName" value="false"/>`	By default, the user name is visible in the **Discovery Robot's** data. To comply with privacy regulations (such as the GDPR) and/or company-issued privacy and security policies, you can hash the user name from all recorded/transmitted data by setting the option to hash (i.e., encrypt) the user name. To hash the username, change the key value to "`true`".
`<add key="standaloneMode" value="false"/>`	In standalone mode, the **Discovery Robot** records user actions while they remain disconnected from the **Discovery Server** • `false` = the **Discovery Robot** is not in standalone mode • `true` = the **Discovery Robot** is in standalone mode **Default Value** `false` For additional configuration of standalone robots, see the section **Standalone Discovery Robots** in *Process Discovery User Guide*.
`<add key="applicationsConfigPath" value="."/>`	Used when `standaloneMode` is set to `True` • Sets the path to the local Applications for Discovery configuration (`blwl.json`) file that defines which applications are recorded • "." indicates same folder Discovery Robot's exe file **Default Value** `"."`
`<add`	Used when `standaloneMode` is set to `True`

Parameter	Description/change
`key="offlineRecordsStoreLimit" value=2000/>`	• Defines the maximum number of records to save locally **Default Value** `2000`
`<add key="EmulatorCommandMaxLength" value="5"/>`	Due to PII and GDPR regulations, all input fields of emulator applications are automatically masked. You can set to unmask up to a configurable number of characters, so they can be interpreted as actions. The available range is 1 to 10. **Default Value** `5`
`<add key="CaptureScreenshots" value="true"/>`	By default, the parameter is set to capture screenshots when recording sessions (`"true"`). To record work-patterns and collect metadata without capturing screenshots, change the key value to `"false"`.
`<add key="UnicodeHookingProvider" value="false"/>`	Sets which hooking provider to use for recording keyboard strokes. Unicode capabilities enable support for multiple language keyboards. • `false` = Use a legacy provider that supports only ASCII • `true` = Use a provider that supports both ASCII and Unicode (for multiple language keyboards) **Default Value** `false`
• `<add key="TlsEnabled" value="false"/>` • `<add key="TlsServer" value=""/>` • `<add key="TlsPort" value="5671"/>`	To configure SSL/TLS on **Discovery Robots**, edit the keys as detailed below: • `y="TlsEnabled" value="true"/>` • `<add key="TlsServer" value="{certificate_server_name}"/>` ◦ `{certificate_server_name}` must be the name exactly as specified on the server certificate file.

Parameter	Description/change
	• `<add key="TlsPort" value="{ssl_listeners_port"/>` ◦ Change this key only if you are using a port other `5671` for SSL/TLS communications. ◦ This should be the same port as you set for `ssl_listeners` (in the `rabbitmq.config` file) when configuring the Discovery Server. **NOTE:** Only SSL/TLS v1.2 is supported.

`<add key="CaptureMethod" value="MIRROR_DRIVER"/>` *(default)*
 	
Other value option:
`"DESKTOP_DUPLICATION"`
`"GDI"`
	The 'CaptureMethod' parameters sets the PDDR capturing method. By default, the capturing method is set to "MIRROR_DRIVER". However, you might need to change the value to "DESKTOP_DUPLICATION" in the following specific case:
	When working via RDP (accessing a remote desktop from your local server) on a remote machine with PDDR and **Windows 10** installed.
	Once you change the key value and save your changes, make sure to restart the PD service.
	In case there is a failure in Mirror driver / desktop duplication - the fallback is GD.

Discovery Robot Installation Logs

The **Discovery Robot** installer records detailed logs of the entire installation process. These logs can be a useful resource for troubleshooting.

To locate the logs:

1. Right-click the **Windows Start** button
2. Select **Run**
3. Type `%temp%`, then hit <ENTER>

   A Windows Explorer window opens to the logged-in user's `Temp` folder, where you can find the logs (look for files with names that start `DiscoveryRobotSetup`)

Upgrading to V. 21.6

About

The procedure described below aims to provide you with all the needed steps to perform an upgrade without losing the recorded sessions of the users.

The procedure described below is applicable when upgrading from Process Discovery version **19.5 or later** to V. 21.6.

Overview

Note: All clients must be upgraded to the latest version as well (21.6)

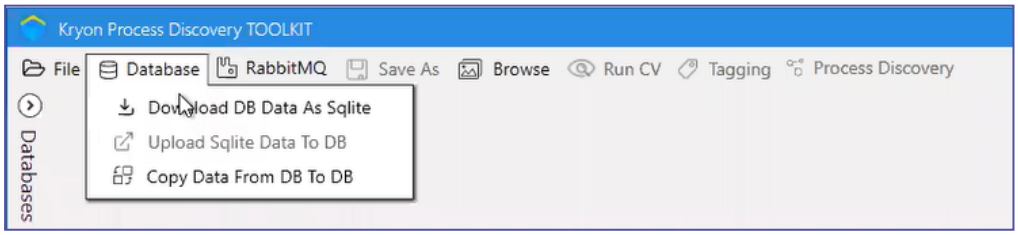
Steps

Step 1: Back-up

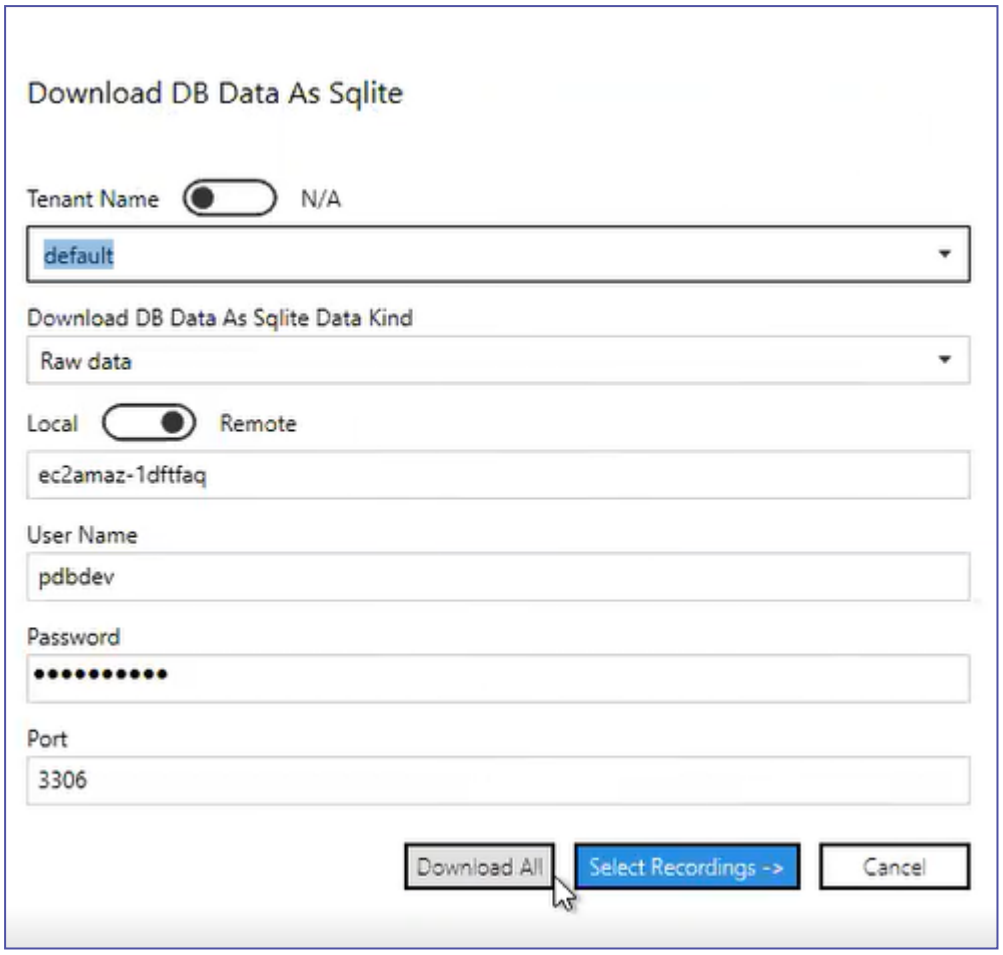
Recordings backup

a. Navigate to the Toolkit folder and open the Toolkit application.

b. From the toolbar, click **Database**.



c. In the **Download DB Data As Sqlite** dialog, select a **Tenant** and click **Download All** or select specific recording to download.

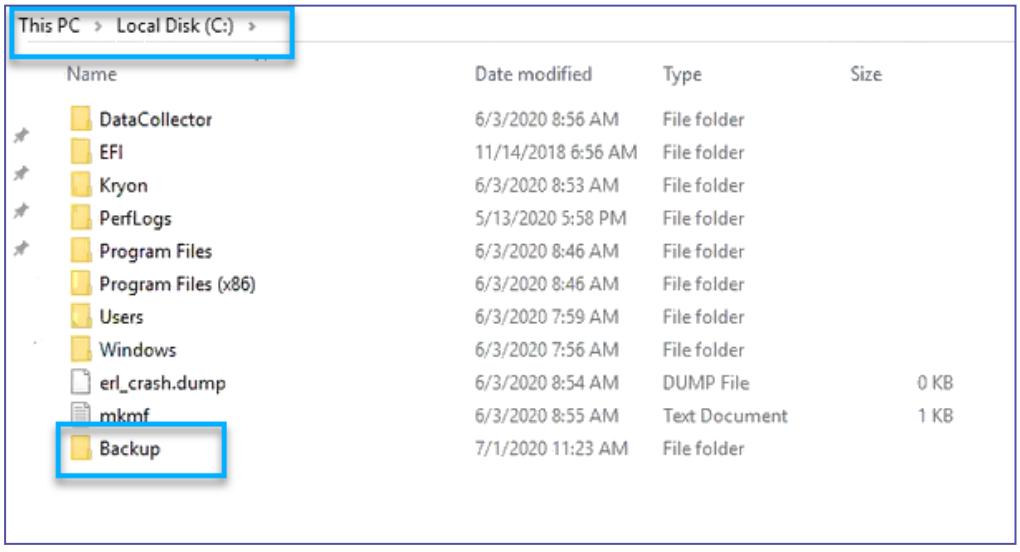


The downloaded recordings are automatically saved under **Toolkit folder > db > RawData**.

Important: Make sure to repeat the procedure for each Tenant (if relevant).

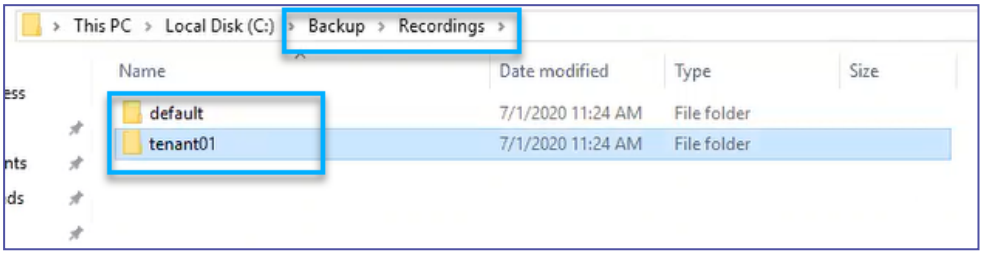
d. Create a new **Backup** folder directly under your default local drive (e.g., **C:** drive.

Important: The newly created backup folder MUST be out of the Kryon folders.



Open the **Backup** folder and:

- Create a sub-folder and name it **Recordings**
- In the **Recordings** folder, create a sub-folder for each Team

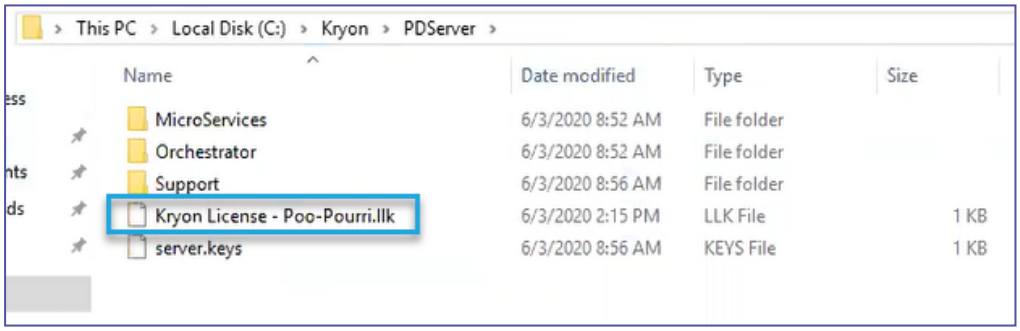


e. Move the downloaded recordings from the **RawData** folder to the relevant Team folder in the **Backup** folder.

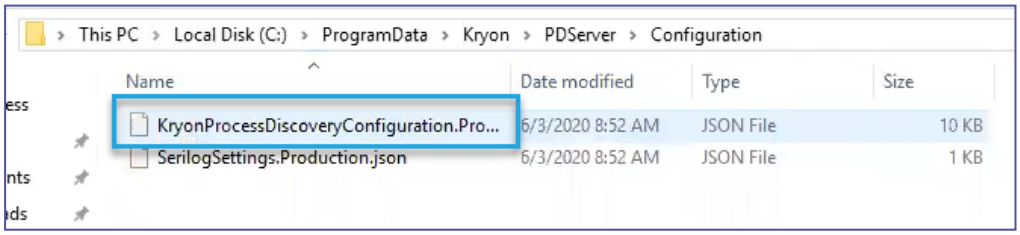
Licenses, configuration, and application definition back-up

Copy-paste the following files directly to the **Backup** folder.

- Kryon License file (found in the **PDServer** folder):



- Kryon Configuration file (found in **PDServer** > **Configuration**):



Backup the application definitions of *all your Teams*:

a. Go to **Settings** > **Applications**.

b. Click **Download application list**.

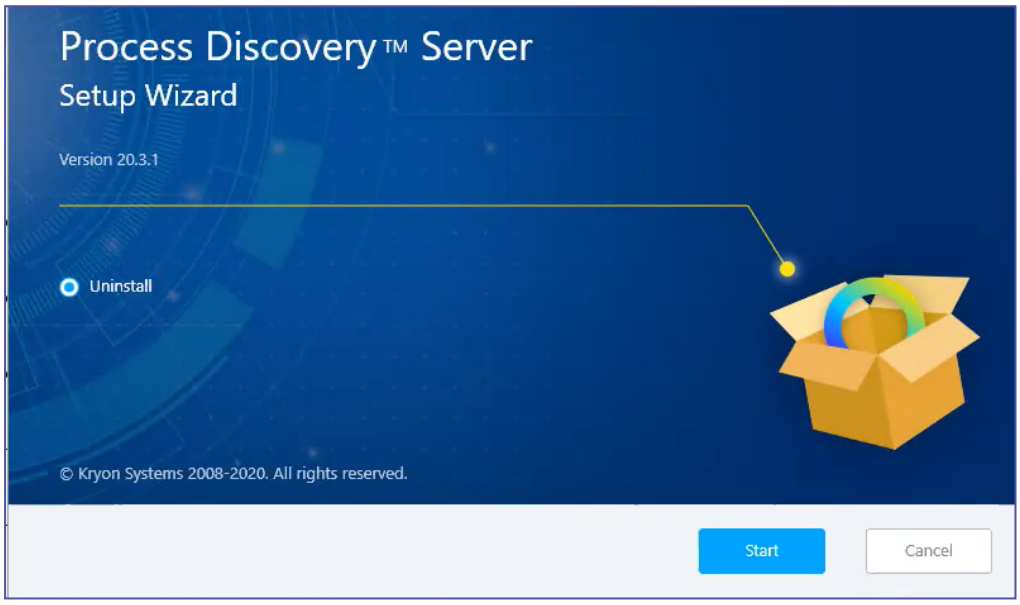
c. Rename the downloaded `.json` file to "`ApplicationsList.json`" and save it to your **Backup** folder.

Step 2: Uninstall the existing Process Discovery

Uninstall the currently installed Process Discovery version like you would normally do.

a. Navigate to **Add or remove programs** > select **Kryon Process Discovery Server** > click **Uninstall**

b. Click **Start** to uninstall. Let the wizard run until completion.



Step 3: Delete the Keycloak server

This step is relevant *only* when upgrading from Process Discovery version 20.1 or later.

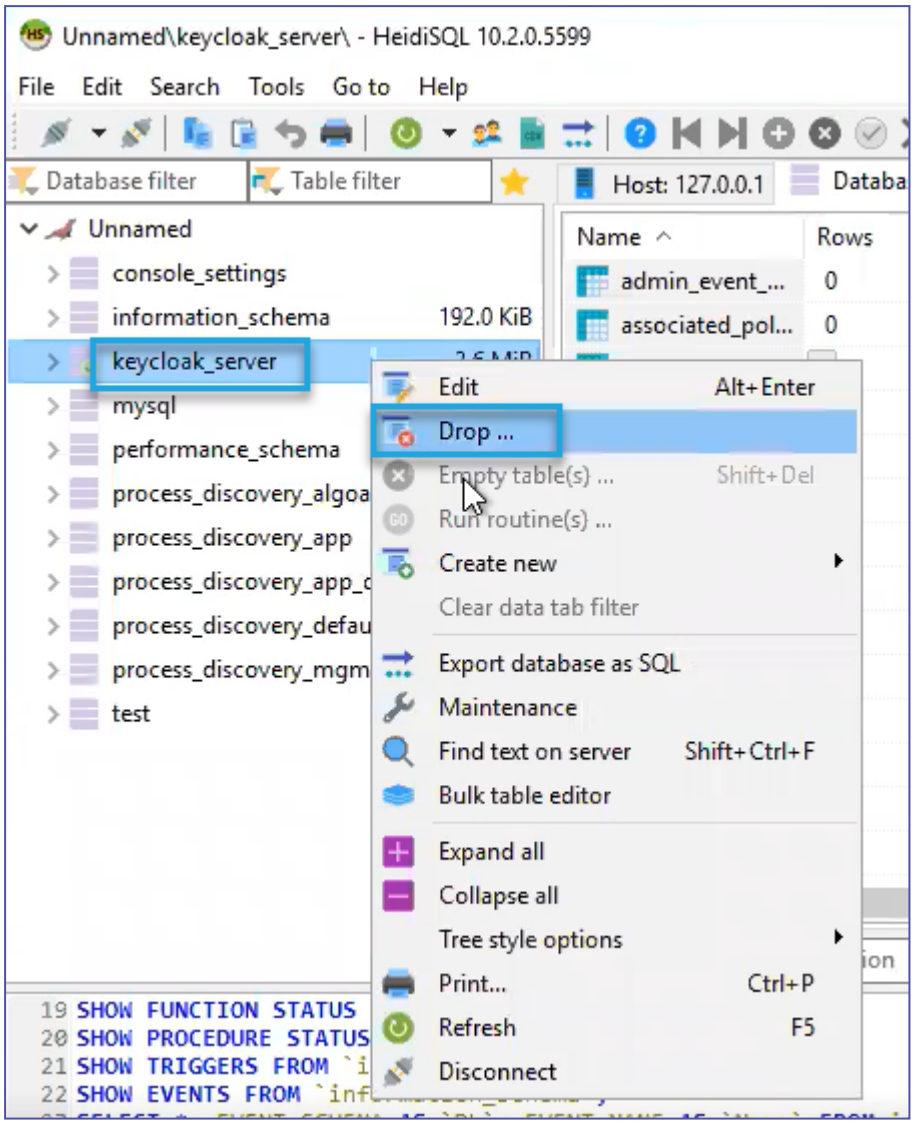
NOTE

The following action deletes the Process Discovery users from Keycloak. The users in Keycloak cannot be backed-up and exported to the latest Process Discovery version (V. 21.6) since the users and login algorithm have been changed and upgraded in version 20.6. You need to re-create the users manually in Process DiscoveryV. 21.6.

a. Open **HeidiSQL**

b. Right-click **keycloack_server**

c. Click **Drop**



Step 4: Delete the database schemes

NOTE

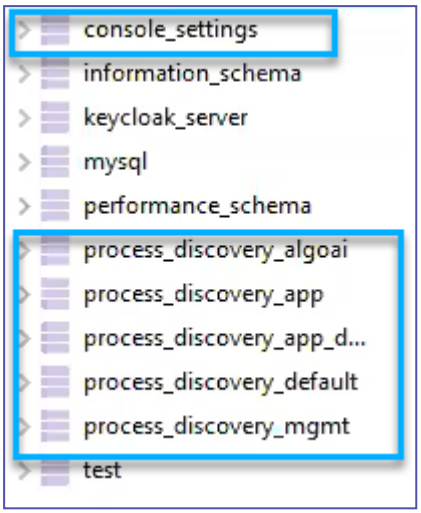
Due to the new changes in the algorithm, you need to re-create the database when upgrading to Process DiscoveryV. 21.6.

a. Open **HeidiSQL**

b. Right-click the e following schemes and then click **Drop**:

- console_settings

- Any schema that starts **process_discovery[...]**



Step 5: Install Process Discovery V. 21.6

Install Process Discovery 20.6 like instructed in the installation section in this guide. When prompted, make sure to select your ***existing and backed-up license file*** and ***RabbitMQ***.

See Installing the Kryon Process Discovery Server.

Step 6: Post-installation configuration

Teams configuration

Re-create your Teams by following the procedure under ***Managing Teams*** (*Process Discovery Console User Guide*).

Recorded sessions import

Import the backed-up recorded-sessions you created in Step 1 by following the procedure under ***Importing (downloading) recorded sessions data*** .

Console users configuration

Re-create your console users by following the procedure under ***Creating Discovery Console Users*** (*Process Discovery User Guide*).

Applications definition configuration

Re-create your applications definition by:

a. Open the back-up file "`ApplicationList`" using a text editor (e.g. Notepad).

b. Refer to the applications definition in the file to re-create the applications in the Process Discovery Console by following the procedure under ***Managing Applications*** (*Process Discovery Console User Guide*).

Silent Installation of Discovery Server

This section describes procedures for a silent installation of the **Discovery Server**:

In this Chapter:

Discovery Server Silent Installation Steps

Follow these steps to do a silent install of the **Discovery Server**:

Step 1: (optional) Change Default Install Folder Location

The format for the location of the **Discovery Server** install folder is `{local drive}:\{InstallFolder}`.

By default the **InstallFolder** location is `{local drive with the most free space}:\Kryon`.

If you choose to change the location of **InstallFolder**:

- It should be targeted to a local folder on a drive with more than 500GB of free disk space
- The `InstallFolder` can't be the root folder of a drive (for example, `C:\` is not allowed)
- The `InstallFolder` folder name can't include spaces (for example, `C:\Install Folder\` is not allowed)
- The `InstallFolder` folder path cannot be longer than 19 characters (20 if a backslash is added at the end)

To change the install folder location:

1. Open the Kryon Process Discovery Server Installation configuration file, `PDServer64BitSetup.exe.json`, with a text editor
2. Enter the value for **InstallFolder** parameter between the corresponding double-quotes, for example:

   `"InstallFolder": "C:\\NewFolder`

   Note: The syntax for specifying folder and file locations in JSON uses a double backslash in each location in which Windows syntax would use a single backslash, for example: `C:\\ProgramFiles\\MariaDB\\`
3. Save the JSON file

Step 2: (optional) Kryon Process Discovery Database Engine

Databases are installed on the **Discovery Server** to collect, analyze, and work with the data received from the robots. Kryon Process Discovery uses four application databases to work its magic:

Database	Database engine
PD application database #1 (data segregated by Team)	MariaDB 10.3.7 (or higher) or MySQL 8.0.11 (or higher)
PD application database #2 (data aggregated for all Teams)	
Process Library database	
User Management platform database	
Raw data collections: Images + metadata recorded by Discovery Robots (segregated by team) collection prefix: `pd-rawdata-${TEAM_GUID}`	MongoDB 4.4.3 (higher)
Masked Images: Only if masking is enables (segregated by team) collection prefix: `pd-masking-rawdata-${TEAM_GUID}`	

The **Discovery Server** installation package automatically installs the database engines.

If you prefer, you can manually install the database engines yourself, or connect to existing ones.

Installation creates the following default users for the databases:

Database	Default User Names
MariaDB	`root` (default pass: Kryon2020! / Kryon123!)
	`pdbdev` (default pass: Kryon2020! / Kryon123! - password can be changed in the installer JSON file)
MongoDB	`kryon-admin` (password is generated during installation and can be fetched from the environment variables)
	`kryon-rw` (default pass: Kryon123!)

Optional steps:

- Configuring a Preexisting Database Engine Installation
- Manually Installing the Database Engines

Step 3: (optional) Update RabbitMQ Permissions

The **Discovery Server** installation package automatically installs RabbitMQ Server.

You can opt to connect to a preexisting RabbitMQ installation, if you have one, later in this installation procedure.

If you want to connect to a preexisting RabbitMQ installation, you may need to add permissions to admin RabbitMQ user at this time. To do so, follow the steps in Add permissions to admin RabbitMQ user for preexisting RabbitMQ version.

Step 4: Open Network Ports

1. The following **default external ports** are opened automatically in the Windows Firewall during the **Discovery Server** installation process. These ports are intended for an installation that *does not use* TLS (see below for SSL/TLS ports). Open them in your hardware firewall prior to installing the server

Default Port (Not using TLS)	Port Used by	Port Used for
5672	Queue Manager (RabbitMQ)	Communication between the **Discovery Robots** and the **Discovery Server**

Default Port (Not using TLS)	Port Used by	Port Used for
80	Console (NGINX)	Communication between the **Discovery Console** and the **Discovery Server**

If you *will* be using TLS, open the following ports in your hardware firewall prior to installing the server.

NOTE: Only SSL/TLS v1.2 is supported.

Default Port (Not using TLS)	Port Used by	Port Used for
5671	Queue Manager (RabbitMQ)	Communication between the **Discovery Robots** and the **Discovery Server**
443	Console (NGINX)	Communication between the **Discovery Console** and the **Discovery Server**

2. You can override the default port setting if you want. At this time, make sure the ports you choose are open in the hardware firewall. After installation you will configure the ports in the server configuration file. Be sure to write down the port numbers you use; you will need them!

(optional) The following *default backend ports* are used in the **Discovery Server**. We recommend *NOT* to open them in the firewall. They are customizable, if required:

Default Port	Port Used for
5058	Communication between the **Discovery Server** and the **User Management Tool**
5001-5004	Communication between the **Discovery Server** and the **Discovery Console**
3306	Internal port for database communications

If you want to customize backend port settings, see:

- Alternative to ports 5001-5004
- Alternative to port 5058
- Alternative to port 3306

Step 5: Copy Installation Files to Local Folder

1. Download the Kryon Process Discovery Server installation files to a local folder on the Server. *Do not copy* these files to the Install Folder location.

- `PDServer64BitSetup.exe`
- `PDServer64BitSetup.exe.json`

2. If you have received your Kryon Process Discovery license file (`.llk`) from your contact at Kryon or the Kryon distribution partner, you can at this point prepare it to be installed automatically at the time of Kryon Process Discovery Server installation. If you wish to install it *after* Discovery Server installation, copy the file to the same folder in which `PDServer64BitSetup.exe` is located

Step 6: Run the Discovery Server Installation Package

1. Access the command prompt and run **as an administrator**
2. Change directory to the location of where you copied the installation files by typing in:

   ```
   CD {Folder location}
   ```

   For example, `CD C:\FolderName\`
3. Hit <ENTER>
4. Type in:

   ```
   START /WAIT PDServer64BitSetup.exe -silent
   ```
5. Hit <ENTER>
6. The installation can take up to 20 minutes, depending on the Server processor power. If needed, the system restarts automatically to complete the install

To verify the discovery server installation

- If the system doesn't require a restart, to verify the installation, immediately after the process completes, run the **ECHO** command with the **%errorlevel%** parameter, by typing: `ECHO %errorlevel%`. The exit codes are shown in the command line.

  The exit codes are as follows:

  - 0 = success
  - 1 = reboot required

- 8 = incompatible machine
- 16 = database failure
- 32 = bad target folder
- 64 = incompatible target drive
- 256 = success with warnings
- 1024 = property value failure
- 2048 = package failure

or

After a system restart during installation, you can verify the installation by viewing the Troubleshooting Discovery Server Installation - Logs.

NOTE

If you wish to install it **after**Kryon Process Discovery Server installation, follow the steps in Step 10: (optional) Install Kryon Process Discovery License After Installation once installation is complete.

Step 7: (optional) Install Kryon Process Discovery License after Installing the Server

Follow these steps to set up your Process Discovery license (if you did not do so previously during installation):

1. Copy the license file you received to the Install Folder
2. From the **Windows Services** app or the **Windows Task Manager > Services** tab:
   - Restart the **Kryon Server -Process Discovery Service**
   - Check the **Status** column of the **Kryon Server -Process DiscoveryService** to ensure that it shows `Running`

NOTE

If you have not yet received your license file, check with your contact at Kryon or the Kryon distribution partner with whom your organization is working.

Step 8: Verify Installation

Verify the **Discovery Server** installation by checking that you can connect to the following components:

NOTE

If you're connecting remotely to the **Discovery Server** or if you defined the server's FQDN when you installed it, enter the FQDN of the server instead of `localhost` in the URLs in the following steps:

1. **Connect to RabbitMQ**

   Log in to RabbbitMQ using the user name `admin` and the default password `Kryon2020!.`

2. **Connect to Process Discovery Console:**

   a. From a web browser on the Discovery Server, enter the following URL: `localhost/console`

   b. Login with the following username and password:

   Username: `Pdconsole`

   Default temporary password: `Pd123456!` (if you changed the password in step 4, use the new password you created)

3. **Connect to SEQ:**

   From a web browser on the Discovery Server, enter the following URL:

   `http://localhost/seq`

4. **Connect to Process Discovery User Management Tool:**

   a. Open an incognito window in Chrome

   Enter the following URL:
   `localhost/auth/admin/kryon/console/#/realms/kryon`

   b. Login with the following username and password:

   Username: `authadmin`

   Default temporary password: `Kryon123!` (you may be asked to change the password. If you do, be sure to write it down)

If you have connected successfully to the above, you're done installing the server!

Next Steps

Install the Discovery Robots

APPENDIX A: Additional Configuration Options

If any of the following scenarios apply to your installation, follow the steps in the relevant topic(s):

In this Chapter:

Manually Installing the Database Engines

MariaDB and MySQL

Databases are installed on the **Discovery Server** to collect, analyze, and work with the data received from the robots. Kryon Process Discovery uses four application databases managed by one or the other database engine:

Database	Database engine
PD application database #1 (data segregated by Team)	MariaDB 10.3.7 (or higher) or MySQL 8.0.11 (or higher)
PD application database #2 (data aggregated for all Teams)	
Process Library database	
User Management platform database	

You can choose manually install MariaDB 10.3.7 (or higher) or MySQL 8.0.11 (or higher).

SeeMariaDB and MySQL Manual Configuration

MongoDB

Kryon Process Discovery uses MongoDB to store user actions in collections:

1. Raw data collections:
   Images + metadata recorded by Discovery Robots (segregated by team)

   Collection prefix - `pd-rawdata-${TEAM_GUID}`
2. Masked images:
   Only if masking is enabled (segregated by team)

   Collection prefix - `pd-masking-rawdata-${TEAM_GUID}`

NOTE

Process Discovery automatically installls MongoDB on the PD installation server. However, you can manually install MongoDB on a remote server and configure the connection.

Connecting to remote MongoDB

You can configure the connetion to MongoDB before or after you install Process Discovery.

Pre-Process Discovery installation configuration

Post-Process Discovery installation configuration

Pre-Process Discovery installation configuration

If you haven't installed Process Discovery yet and you wish to configure MongoDB first, then follow these instructions:

1. Download, install, and configure MongoDB and Robo 3T (optional) following the instructions in MongoDB Manual Configuration .
2. Open the file `Kryon_PDServer64BitSetup.exe.json` and set the parameters according to **New Value** column in the below table. Note that some parameters are optional.
3. Run the Process Discovery server installation package.

Parameter	Description	Default value	New Value	Comments
INSTALL_MONGODB	Whether to install MongoDB on Process Discovery server	true	false	-
MONGODB_CONNECTION_STRING	MongoDB connection	mongodb://localhost:27017/kryon-db?authSource=admin	mongodb://**${MONGO_SERVER_FQDN}**:27017/kryon-db?authSource=admin	Edit the value to include the remote server FQDN. You can also change the default port (27017) and the database name.
MONGODB_DBNAME	Database name for Process Discovery usage	kryon-db	-	This is an optional configuration, if you choose to

				change the DB name, make sure you change the connection string appropriately.
MONGODB_ADMIN_USER	Admin user name	kryon-admin	-	Optional configuration
MONGODB_ADMIN_USER_AUTH	Admin user password	Kryon2021!	-	Optional configuration
MONGODB_KRYON_USER	Default user name	kryon-rw	-	Optional configuration
MONGODB_KRYON_USER_AUTH	Default user password	Kryon123!	-	Optional configuration

Post-Process Discovery installation configuration

If you have already installed Process Discovery and you wish to configure MongoDB on a remote server, then follow these instructions:

1. Download, install, and configure MongoDB and Robo 3T (optional) following the instructions in MongoDB Manual Configuration .
2. Open the file `${Installation_Folder}\Kryon\installer-assets\config\prod\scripts\config.prod.properties.json` and change the following parameters:

Parameter	Description	Default value	New Value	Comments
INSTALL_MONGODB	Whether to install MongoDB on Process Discovery server	true	false	-
MONGODB_CONNECTION_STRING	MongoDB connection	mongodb://localhost:27017/kryon-db?authSource=admin	mongodb://**${MONGO_SERVER_FQDN}**:27017/kryon-db?authSource=admin	Edit the value to include the remote server FQDN. You can also change the default port (27017) and the database name.
MONGODB_DBNAME	Database name for Process Discovery usage	kryon-db	-	This is an optional configuration, if you choose to change the DB name,

				make sure you change the connection string appropriately.
MONGODB_ADMIN_USER	Admin user name	kryon-admin	-	Optional configuration
MONGODB_KRYON_USER	Default user name	kryon-rw	-	Optional configuration

3.

Run `configureAll.ps1` by opening CMD as administrator and running:

```
 CD C:\Kryon\installer-assets\config\prod\scripts
powershell.exe -Command "C:\Kryon\installer-
assets\config\prod\scripts\configureAll.ps1 -h 'C:\Kryon' -configDir
'C:\Kryon\config' -n prod -servicesDir 'C:\Kryon\PDServer\MicroServices' -
utilsDir 'C:\Kryon\PDServer\Support'"
```

4. Change users passwords (optional):
   a. Open System Environment variables (Start > Edit system environment variables > Environment Variables)
   b. Edit the values of `MONGODB_ADMIN_USER_AUTH` or `MONGODB_KRYON_USER_AUTH`
   c. Click OK/Save.
5. Restart the "Kryon Server - Process Discovery Service" Windows service

MariaDB and MySQL Manual Configuration

MariaDB - Manual Installation

NOTE

The Administrative user needs full permissions to create schema and tables.

Kryon Process Discovery supports the *Windows version* of MariaDB 10.3.7 (or higher). If you are installing MariaDB manually, you can download the software from here: https://downloads.mariadb.org/.

Process Discovery uses two hard coded usernames:

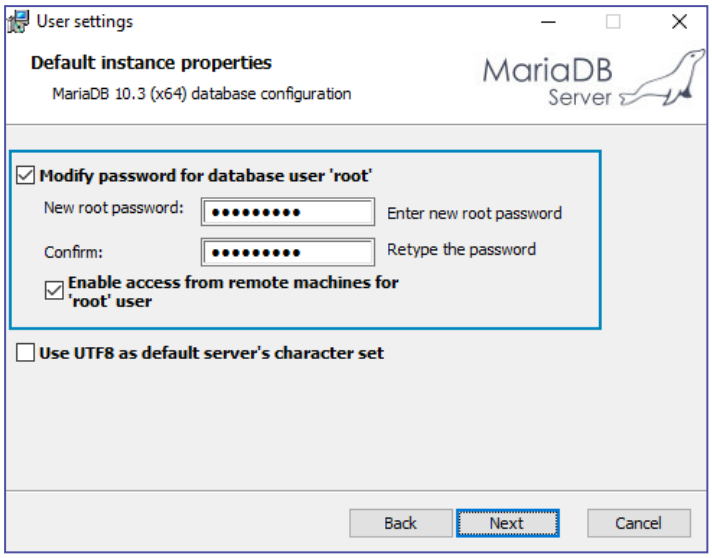
- `root`: The Root username for adminatrative logins
- `pdbdev`: the Pdbdev username for viewing the database

You should use the default options for each screen in the MariaDB installation wizard, with the following important exceptions:

ROOT PASSWORD

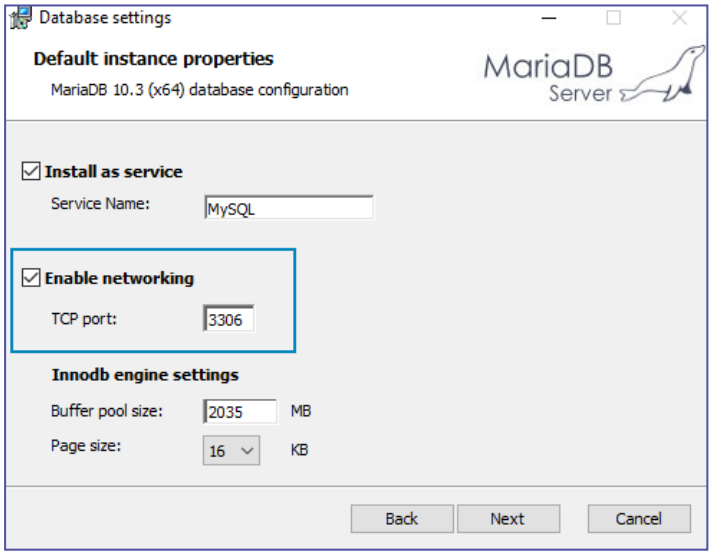
1. Select the option to modify the root password, and enter and make note of the changed password
   - If you want the Kryon Process Discovery Server installation package to use the default password value, enter `Kryon2020!` as the password
   - If you choose to use a different password here, *be sure to make note of it*.
2. Be sure to select the option to **enable access from remote machines for 'root'**

user



TCP PORT

1. You can choose to change the default TCP port – **3306**– if you wish. But if you do so, *be sure to make note of it.*



2. Complete the installation using the default options for each screen in the MariaDB installation wizard.
3. If you are doing a regular installation, you're done! For a silent installation, continue on.

Configure the database configuration file (recommended)

1. Open the database configuration file `my.ini` and set the following parameter values to improve database performance:

```
long_query_time=1
max_connections=200
table_open_cache=1000
tmp_table_size=64M
thread_cache_size=200
key_buffer_size=128M
innodb_flush_log_at_trx_commit=1
innodb_log_buffer_size=16M
innodb_buffer_pool_size=8G
innodb_log_file_size=1000M
innodb_thread_concurrency=8
innodb_buffer_pool_instances=8
innodb_open_files=2000
innodb_stats_on_metadata=OFF
innodb_checksum_algorithm=crc32
back_log=200
max_allowed_packet=1G
table_definition_cache=500
character_set_server=utf8mb4
collation_server=utf8mb4_unicode_ci
```

2. Save the file
3. Restart the database service

Update the installation configuration file (for silent installation)

For a silent installation, change the following values in the JSON file provided with the installation package, **PDServer64BitSetup.exe.json**:

1. Open the JSON file provided with the installation package, **PDServer64BitSetup.exe.json** with a text editor

2. Edit the MariaDB `root` password : change the value of the `DB_ROOT_PASSWORD` parameter with the password you gave for the `root` above
3. Installation creates an additional application user, `Pdbdev` with default password `Kryon2020!`. If you want to change this user name and password:

   (optional) Edit the value of the `DB_USER_NAME` with your preferred user name and edit the value of the `DB_USER_PASSWORD` parameter with a new password
4. Save the JSON file. You're done!

MySQL - Manual installation

If you choose to use MySQL for the application databases, it must always be manually installed prior to Discovery Server installation.

Process Discovery uses two hard coded usernames:

- `root`: The Root username for adminatrative logins
- `pdbdev`: the Pdbdev username for viewing the database

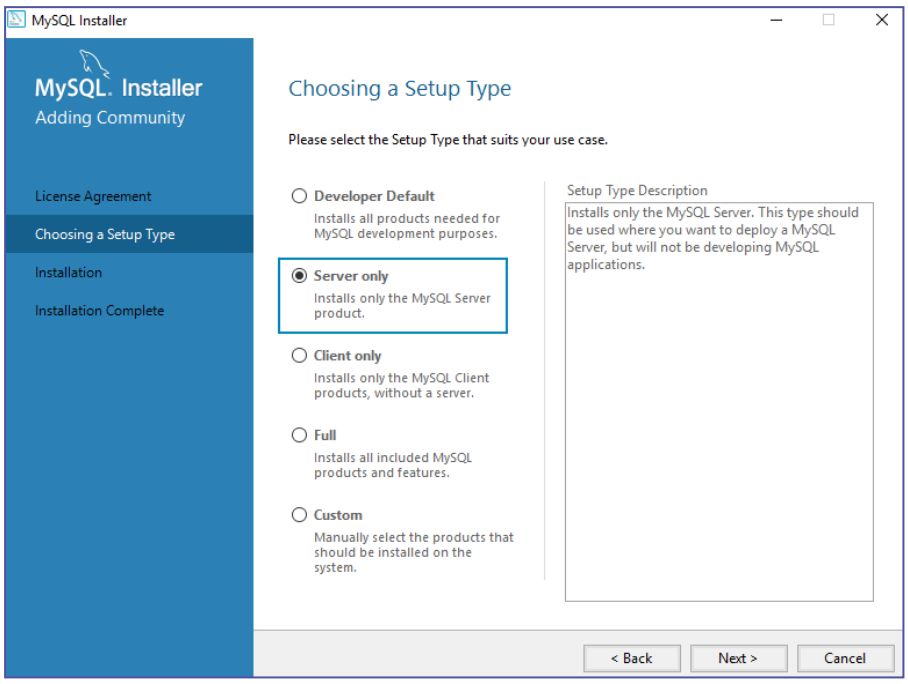
NOTE

The Administrative user needs full permissions to create schema and tables.

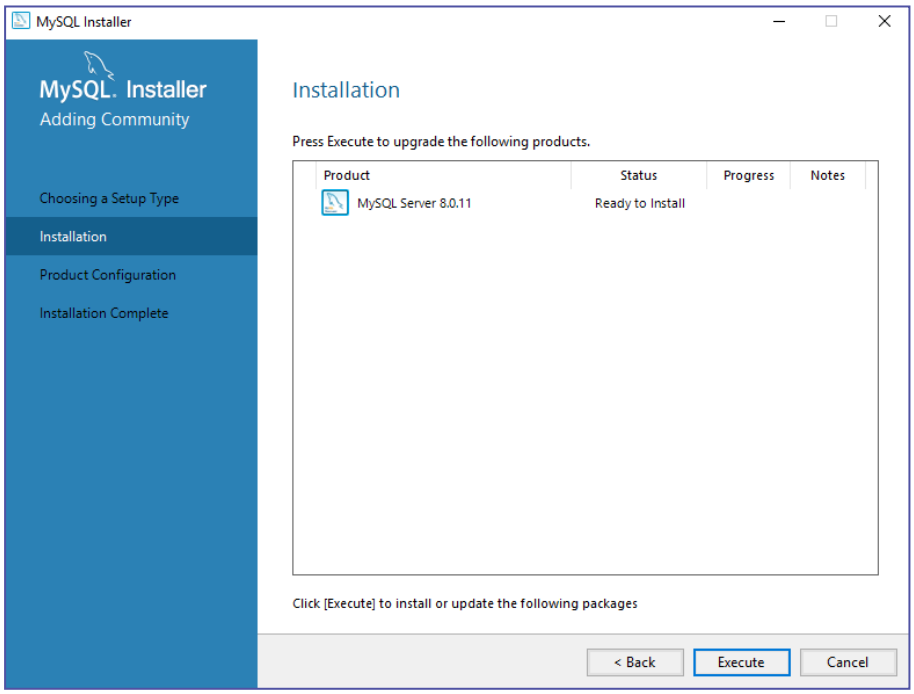
Run the installation package, and install with the following options:

Choosing a Setup Type

1. Select **Server only**, then click the **Next >** button



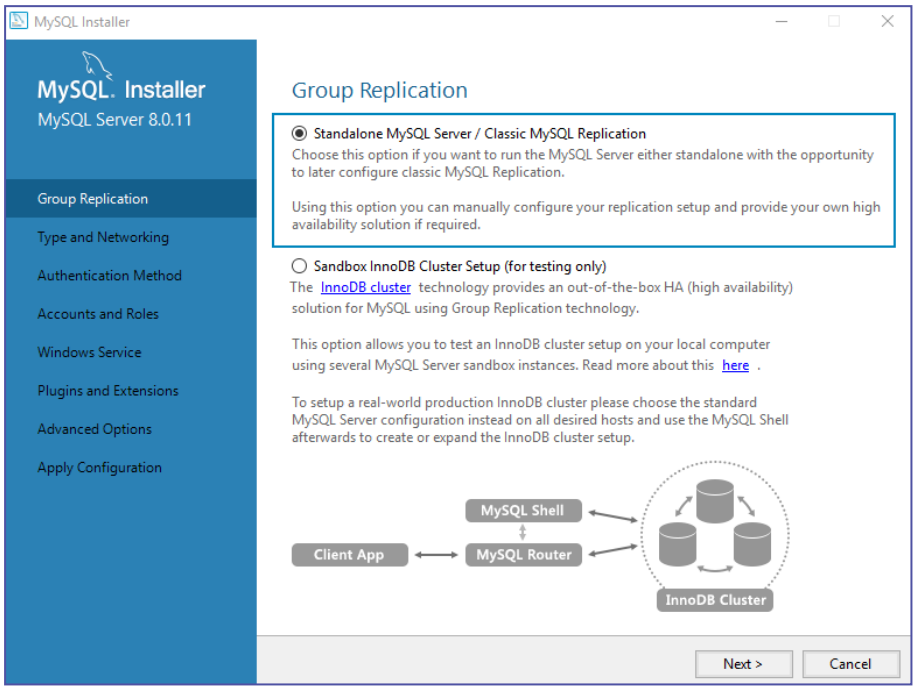
2. On the next screen, click the **Execute** button, and the package will be installed



3. You will then be prompted to configure the package

Group Replication

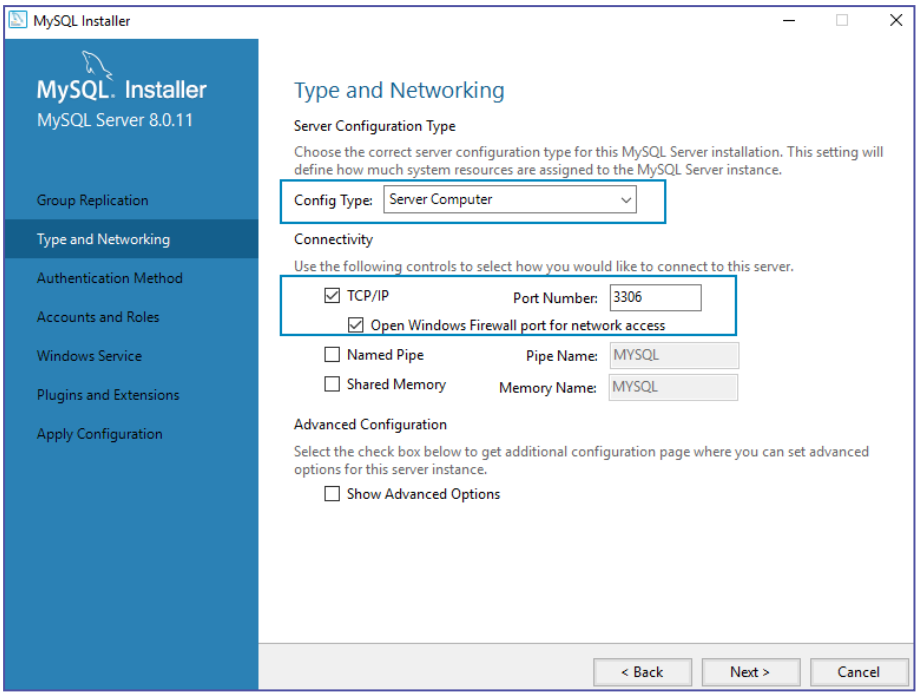
- Select **Standalone MySQL Server / Classic MySQL Replication**



Type and Networking

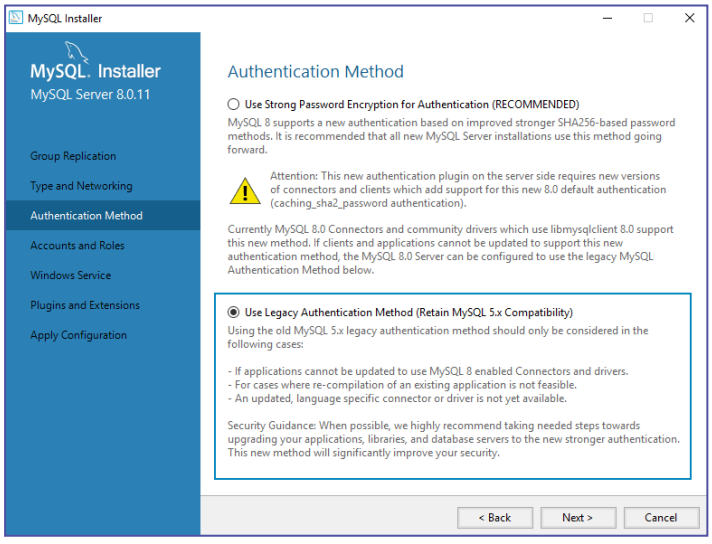
1. From the **Config Type** dropdown list, select **Server Computer**
2. In the **Connectivity** section:
   - Select the option for **TCP/IP**
   - Select the option to **Open Windows Firewall port for network access**
   - **Port Number:** You can choose to change the default Port Number– **3306**– if you wish. But if you do so, ***be sure to make note of it***. You will need to specify

it as an installation parameter during **Discovery Server** installation.

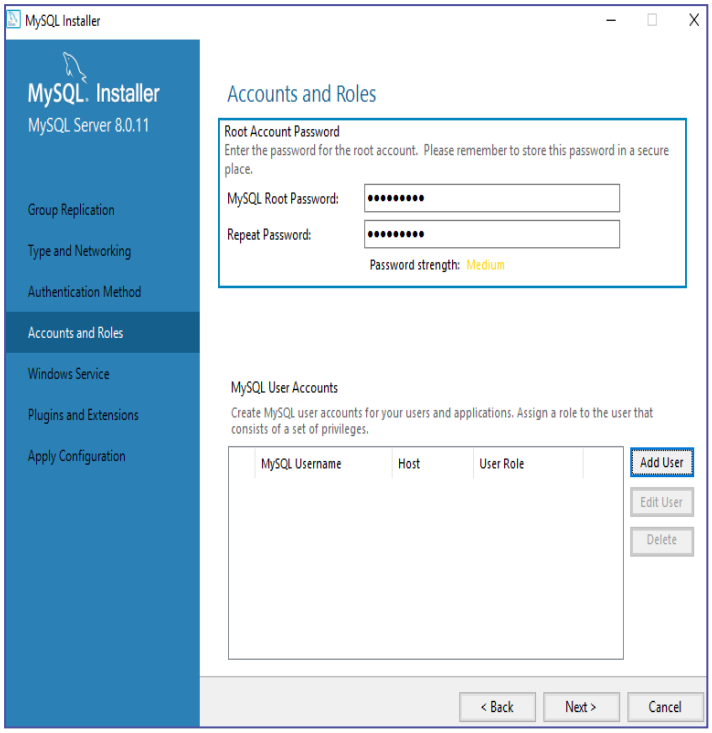


Authentication method

- Select **Use Legacy Authentication Method (Retain MySQL 5.x Compatibility)**

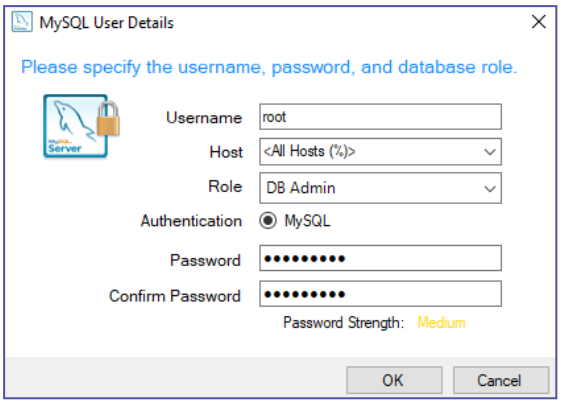


Accounts and Roles



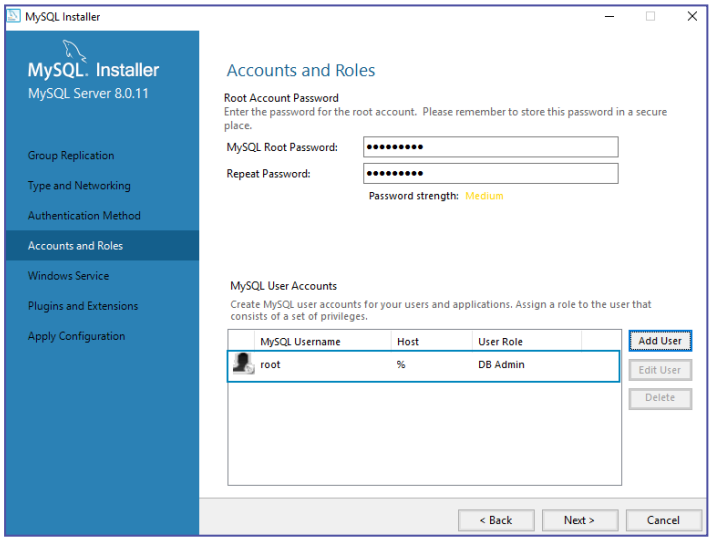
1. In the **Root Account Password** section, enter and make note of the changed password
   - If you want the Discovery Server installation package to use the default password value, enter `Kryon2020!` as the password
   - If you choose to use a different password here, *be sure to make note of it*. You will need to specify it as an installation parameter during **Discovery Server** installation.
2. In the **MySQL User Accounts** section, click the **Add User** button, and create a user with the following properties:
   - **Username:** root
   - **Host:** <All Hosts (%)>
   - **Role:** DB Admin
   - **Authentication:** MySQL

- **Password:** same password as you entered in step 1 above



> **NOTE**
>
> The Administrative user needs full permissions to create schema and tables.

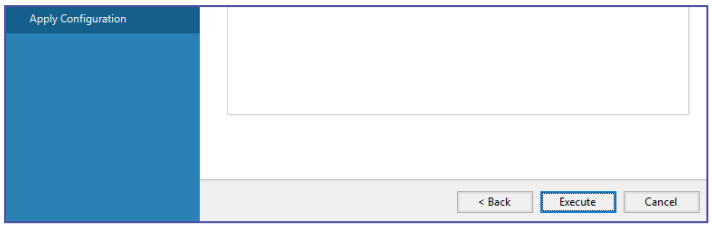
3. Then click the **OK** button to return to the **Accounts and Roles** screen, which should look like this after the user has been created:



4. Repeat steps 2 - 3, creating an additional user `pdbdev`

Apply configuration

- At the end of the configuration process, you will be prompted to apply the configuration. Click the **Execute** button, and you're done installing MySQL!


Apply Configuration
< Back
Execute
Cancel

Configure the database configuration file

1. Open the database configuration file `my.ini`, and add the following line:

```
# Set default time zone
default-time-zone = '+00:00'
```

   `default-time-zone='timezone'` where 'timezone' represents the UTC time zone of the server, for example, `default-time-zone='+00:00'` (the format `'system'` is not supported)

2. We recommend the following settings to improve database performance. Set the following parameter values in `my.ini`:

```
long_query_time=1
max_connections=200
table_open_cache=1000
tmp_table_size=64M
thread_cache_size=200
key_buffer_size=128M
innodb_flush_log_at_trx_commit=1
innodb_log_buffer_size=16M
innodb_buffer_pool_size=8G
innodb_log_file_size=1000M
innodb_thread_concurrency=8
innodb_buffer_pool_instances=8
innodb_open_files=2000
innodb_stats_on_metadata=OFF
innodb_checksum_algorithm=crc32
back_log=200
max_allowed_packet=1G
table_definition_cache=500
character_set_server=utf8mb4
collation_server=utf8mb4_unicode_ci
```

3. Save the file
4. Restart the database service

MongoDB Manual Configuration

Manually install MongoDB 4.4.3 (or higher) by following these steps:

1. Download MongoDB from here
2. Create the `kryon-db` database
3. Create the Process Discovery Administrator and default users following one of the options below:

   Option 1: Create users through running designated commands

   Option 2: Install and configure Robo 3T to create database users
4. Add TLS configuration for MongoDB (optional)

Create the `kryon-db` database

a. Create a new cluster of MongoDB with engine Version 4.0.0 and only one instance within the database subnet

IMPORTANT

Make sure to activate encryption and audit logs

b. Connect to MongoDB.
You can use Robo 3T (`{InstallFolder}\Kryon\SupportTools\mongodb-client-robo3t-1.4.1-windows-x86_64-122dbd9\robo3t-1.4.1-windows-x86_64-122dbd9\robo3t.exe` to connect to MongoDB.

c. Create the database `kryon-db`

d. Execute the following query on the MongoDB to create the users (or use Robo 3T to create users) :

```
```json
use admin
db.createUser(
{
user: "kryon-admin",
pwd: "<<new-password-for-mongodb>>",
roles: [ "root" ]
}
)
db.createUser(
{
user: "kryon-rw",
```
```

```
pwd: "<<new-password-for-mongodb>>",
roles: [ { role : "readWrite", db: "kryon-db"}]
}
)
```

Create users through running designated command

Run the following commands to create users:

a. To create Process Discovery Administrator user run:

```
cd C:\Kryon\MongoDB\Server\4.4\bin
mongo
use admin
db.createUser( { user: "kryon-admin", pwd: "Kryon2021!", roles: [ 'root'] } )
```

b. To create Process Discovery default user run:

```
cd C:\Kryon\MongoDB\Server\4.4\bin
mongo
use admin
createUser({user: 'kryon-rw', pwd: 'Kryon123!', roles: [ { role :
'readWrite', db: 'kryon-db'}]})
```

Add TLS configuration for MongoDB (optional)

a. Copy the file `rds-combined-ca-bundle.pem` to `C:\Kryon\config\prod\general\ssl\rds-combined-ca-bundle.pem` (you might need to create the folder first)

b. Replace the content of the file `C:\Kryon\config\prod\general\mongodb-connector-kryon-db-r.json` with the following ***and*** insert the domain:

```
```json
{
"name" : "mongodb-connector-kryon-db-r",
"value" : {
"dbName" : "kryon-db",
"url" : "mongodb://<<domain-of-cluster>>:27017/kryon-
db?authSource=admin&tls=true&tlsCAFile=rds-combined-ca-
bundle.pem&replicaSet=rs0&readPreference=secondaryPreferred&retryWrites=false"
,
"debug" : true,
"options": {
"poolSize": 10,
"tls": true,
"tlsInsecure": false,
"tlsCAFile": "rds-combined-ca-bundle.pem",
"useNewUrlParser": true,
"sslValidate": true
```
```

```
}
}
}
```

c. Replace the content of the file `C:\Kryon\config\prod\general\mongodb-connector-kryon-db.json` with the following ***and*** insert the password and the domain:

```
```json
{
"name": "mongodb-connector-kryon",
"value": {
"description": "",
"dbName" : "kryon-db",
"url": "mongodb://kryon-rw:<<password>>@<<domain-of-cluster>>:27017/kryon-db?authSource=admin&tls=true&tlsCAFile=rds-combined-ca-bundle.pem&replicaSet=rs0&readPreference=secondaryPreferred&retryWrites=false",
"debug": true,
"options": {
"poolSize": 10,
"tls": true,
"tlsInsecure": false,
"tlsCAFile": "rds-combined-ca-bundle.pem",
"useNewUrlParser": true,
"sslValidate": true
}
}
}
```
```

d. Replace the content of the file `C:\Kryon\config\prod\general\mongodb-connector-kryon-db-admin.json` with the following ***and*** insert the password and the domain:

```
```json
{
"name": "mongodb-connector-kryon",
"value": {
"description": "",
"dbName" : "admin",
"url": "mongodb://kryon-admin:<<password>>@<<domain-of-cluster>>:27017/admin?authSource=admin&tls=true&tlsCAFile=rds-combined-ca-bundle.pem&replicaSet=rs0&readPreference=secondaryPreferred&retryWrites=false",
"debug": true,
"options": {
"poolSize": 10,
"tls": true,
"tlsInsecure": false,
"tlsCAFile": "rds-combined-ca-bundle.pem",
"useNewUrlParser": true,
"sslValidate": true
}
```
```

```
}
}
```

e. Open CMD and run:

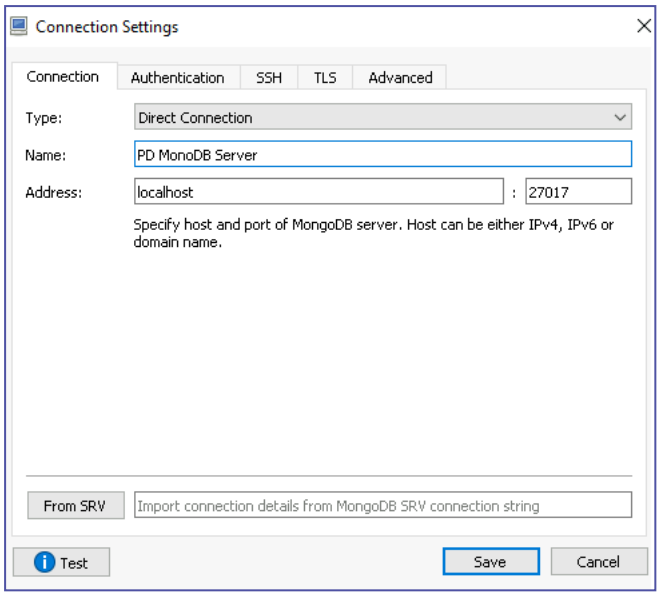
```bat
cd C:\Kryon\PDServer\Support\kryon-admin-cli
set CONFIG_DIR=C:\Kryon\config
set NODE_ENV=prod
set KRYON_ENC_CFG=C:\Kryon\config\prod\general\kryon-decrypt.json
set NODE_PATH=C:\Kryon\PDServer\MicroServices\node_modules
node bin\cli.js --provider=mongodb --command=uploadCodex --
collectionName=masking-codex --
fileName=C:\Kryon\PDServer\MicroServices\AlgoNextGen\masking-private-codex.csv
--tenantName=default
```

Install and configure Robo 3T to create database users

Robo 3T (Robomongo) is the free, lightweight, open-source MongoDB GUI with an embedded mongo shell.

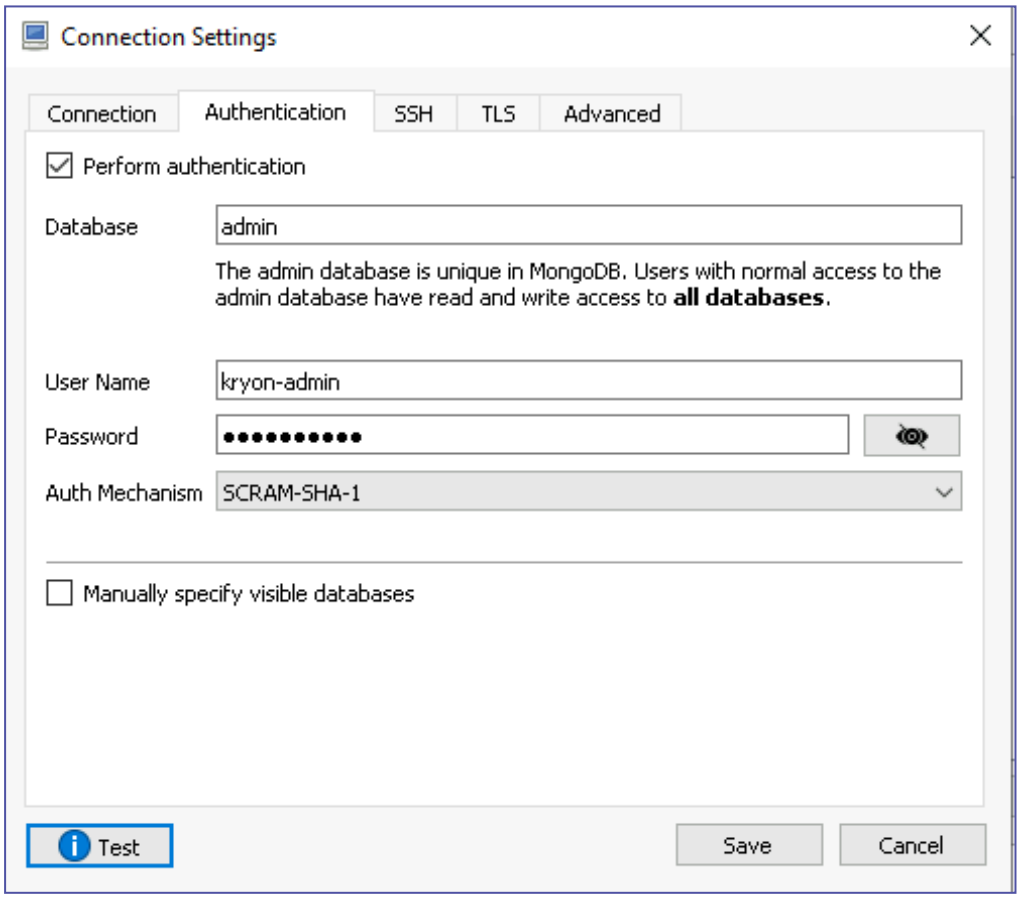
1. Downland and install Robo 3T:Floating Content Panels
   - Robo 3T is included in Process Discovery installation under `${InstallationFolder}\Kryon\SupportTools`.
   - You can download Robo 3T also from here.
2. Open Robo 3T > Click on **File** > Click on **Connect**
3. Click **Create** to create a new connection
4. In the connection tab, specify the MongoDB server FQDN and port

   EXAMPLE:

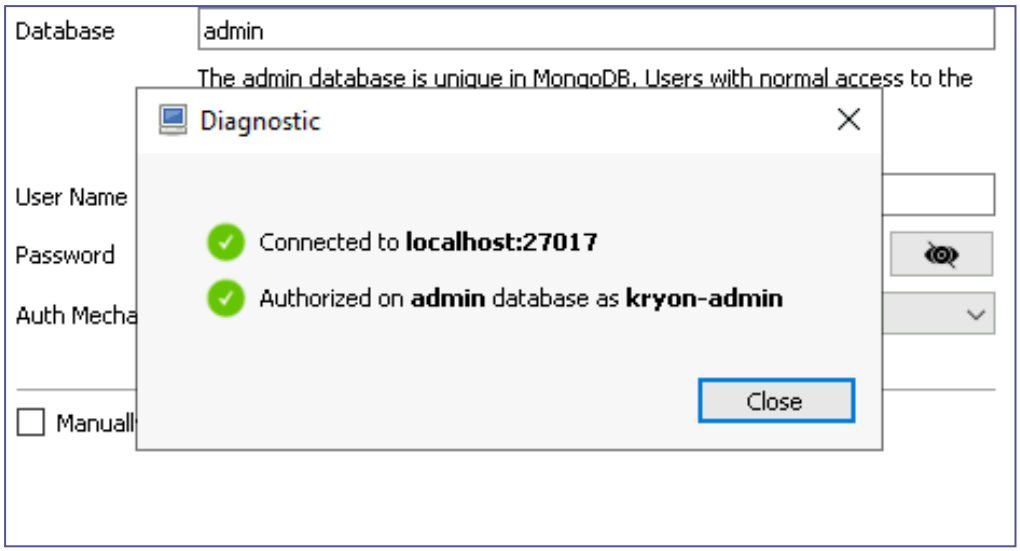


5. In the Authentication tab, Check the "Perform authentication" check-box and specify the username and password

EXAMPLE:



6. Make sure the details are accurate by clicking on **Test**



7. Click **Save**
8. Select the newly created connection and click **Connect**.

Configuring a Preexisting Database Engine Installation

⟵ *to return to silent installation instructions*

Kryon Process Discovery supports using a dedicated **Windows version** of MariaDB 10.3.7 (or higher) and MySQL 8.0.11 (or higher). If you already have MariaDB or MySQL installed on your server, follow the steps below to configure database installation.

Preexisting database engine configurations must always be done prior to Discovery Server installation.

Configure the database configuration file

1. For MySQL, open the database configuration file `my.ini`, and add the following line:

```
# Set default time zone
default-time-zone = '+00:00'
```

   `default-time-zone='timezone'` where 'timezone' represents the UTC time zone of the server, for example, `default-time-zone='+00:00'` (the format `'system'` is not supported)

2. For both MySQL and MariaDB, we recommend the following settings to improve database performance. Set the following parameter values in `my.ini`:

```
long_query_time=1
max_connections=200
table_open_cache=1000
tmp_table_size=64M
thread_cache_size=200
key_buffer_size=128M
innodb_flush_log_at_trx_commit=1
innodb_log_buffer_size=16M
innodb_buffer_pool_size=8G
innodb_log_file_size=1000M
innodb_thread_concurrency=8
innodb_buffer_pool_instances=8
```

```
innodb_open_files=2000
innodb_stats_on_metadata=OFF
innodb_checksum_algorithm=crc32
back_log=200
max_allowed_packet=1G
table_definition_cache=500
character_set_server=utf8mb4
collation_server=utf8mb4_unicode_ci
```

3. Save the file
4. Restart the database service

Configure the discovery server json configuration file parameters

See here for a full explanation of Discovery Server JSON configuration file parameters.

1. Open the JSON file provided with the installation package, **PDServer64BitSetup.exe.json** with a text editor
2. Enter the values for your existing database engine (between the double-quotes that correspond to the parameters below):

```
"DB_SERVICE_NAME": "your service name"
"DB_SERVER": "your DB server"
"DB_PORT": "your DB port"
"DB_ROOT_NAME": "DB root user name"
"DB_ROOT_PASSWORD": "DB root user password"
"DB_USER_NAME": "DB user name"
"DB_USER_PASSWORD": "DB user password"
```

3. Save the JSON file

Configuring the Event Log File Structure

You can combine **Kryon Process Discovery** with Process Mining techniques by downloading an Event Log (in csv format) to apply to an external process mining tool.

By default, the Event Log's file structure uses comma-separated values. Some process mining tools require semicolon delimiters instead.

You can configure the Event Log file structure to use semicolon delimiters, as follows:

1. Go to `[Installation Folder]/Kryon/PD Server/Micro Services/Kryon.DocManager/` and open the file **`appsetting.Production.json`**.
2. Change the value of the `UseCsvSemicolonDelimiter` parameter from false to **true**.

Process Discovery over HTTPS

TLS is an encryption protocol intended to keep data secure when transferred over a network. This guide describes steps required to ensure that the Process Discovery server uses TLS protocol.

In this Chapter:

Prerequisites

- Process Discovery v21.4 (and up) installed (Server and Clients)
- SSL Certificate = Server certificate (*.crt format) + Server key file (*.pem format) + Ca bundle certificate (*.pem format)

Process Discovery TLS Configuration

NOTE

The system file paths in this guide are the default Process Discovery Server installation

Steps Overview

Step 1 - Stop Process Discovery services

Shutdown the following Kryon windows services:

- Kryon Server - Authentication Gateway
- Kryon Server - Authentication Server
- Kryon Server - Process Discovery Service

Step 2 - Configure TLS in Aerboase

a. Copy the server `.crt` and `.key` certificate files to Aerobase SSL folder (`C:\Kryon\IDP\Aerobase\Configuration\ssl`)

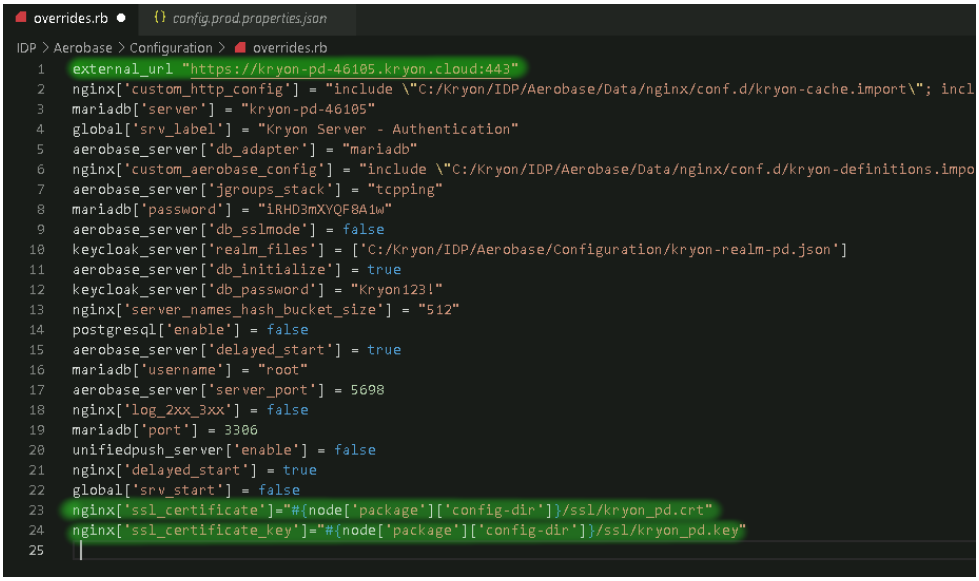
b. Edit the Aerobase configuration file `C:\Kryon\IDP\Aerobase\Configuration\overrides.rb` as follwing:

- Modify the property `external_url`: Change the protocol to `https` and the port to `443`
- Insert the following properties and set the file names to be exactly like the copied certificate names:
  - **nginx['ssl_certificate']=**"`#{node['package']['config-dir']}/ssl/<FILE_NAME>.crt`"
  - **nginx['ssl_certificate_key']=**"`#{node['package']['config-dir']}/ssl/<FILE_NAME>.key`"

IMPORTANT

Use only slashes "/" in the files path (not backslash! "\").

EXAMPLE:



c. Run Aerobase configure command:

```
powershell
(Invoke-Command {cmd.exe /c C:\Kryon\IDP\Aerobase\Aerobase\bin\aerobase-ctl.bat reconfigure} | Out-File aerobase-reconfigure-output.txt)
```

d. Validate the Aerobase TLS configuration by reviewing the file:

`C:\Kryon\IDP\Aerobase\Data\nginx\conf\`**`aerobase-http.conf`**

EXAMPLE:

overrides.rb ● aerobase-http.conf ×

IDP > Aerobase > Data > nginx > conf > aerobase-http.conf

```
9   ####################################
10  ##         configuration          ##
11  ####################################
12
13  upstream softwarebl {
14    zone upstream_dynamic 64k;
15    server 127.0.0.1:5698;
16  }
17
18
19  server {
20    listen *:443 ssl;
21    server_name kryon-pd-46105.kryon.cloud;
22    server_tokens off; ## Don't show the nginx version number, a security best practice
23
24    ## Increase this if you want to upload large attachments
25    ## Or if you want to accept large git objects over http
26    client_max_body_size 250m;
27
28    ## Strong SSL Security
29    ## https://raymii.org/s/tutorials/Strong_SSL_Security_On_nginx.html & https://cipherli.st/
30    ssl on;
31    ssl_certificate C:/Kryon/IDP/Aerobase/Configuration/ssl/kryon_pd.crt;
32    ssl_certificate_key C:/Kryon/IDP/Aerobase/Configuration/ssl/kryon_pd.key;
33
34    # Unifiedpush needs backwards compatible ciphers to retain compatibility with Java IDEs
35    ssl_ciphers 'ECDHE-RSA-AES256-GCM-SHA384:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-RSA-AES256-SHA384:E
36    ssl_protocols  TLSv1 TLSv1.1 TLSv1.2;
37    ssl_prefer_server_ciphers on;
38    ssl_session_cache  builtin:1000  shared:SSL:10m;
39    ssl_session_timeout  5m;
40
41
42    ## Individual nginx logs for this unifiedpush vhost
43    access_log  logs/aerobase_access.log combined if=$loggable;
44    error_log   logs/aerobase_error.log;
45
46    include "C:/Kryon/IDP/Aerobase/Data/nginx/conf.d/kryon-definitions.import"; include "C:/Kryon/
47    include aerobase-locations.import;
48  }
49
```

IMPORTANT

- If an issue occurs, check aerobase and nginx logs: `C:\Kryon\IDP\Aerobase\Logs`
- Upon any change to the aerobase override.rb file, make sure to re-run aerobase configuration
- The file names of the certificate *must not* contain dot characters (e.g., my.cert.crt)

Step 3 - Configure Process Discovery

a. Modify the Process Discovery configuration by editing the file `C:\Kryon\installer-assets\config\prod\scripts\config.prod.properties.json` and setting the following values:

- "HTTP_TYPE": "**https**"
- "NGINX_PORT": **443**
- "NODEJS_CA_CERTS": "**<PATH_TO_CA_BUNDLE>\ca_bundle.pem**"

EXAMPLE:

```
"NODEJS_CA_CERTS":
"C:\\Kryon\\IDP\\Aerobase\\Configuration\\ssl\\cert.pem",
```

b. Run Process Discovery configureAll command:

```
CD C:\Kryon\installer-assets\config\prod\scripts
powershell.exe -Command "C:\Kryon\installer-
assets\config\prod\scripts\configureAll.ps1 -h 'C:\Kryon' -configDir
'C:\Kryon\config' -n prod -servicesDir 'C:\Kryon\PDServer\MicroServices' -
utilsDir 'C:\Kryon\PDServer\Support'"
```

c. Modify the Console configuration by editing the file `C:\Kryon\Console\Web\ConsoleX\assets\data\`**`appConfig.prod.json`** and setting the following values:

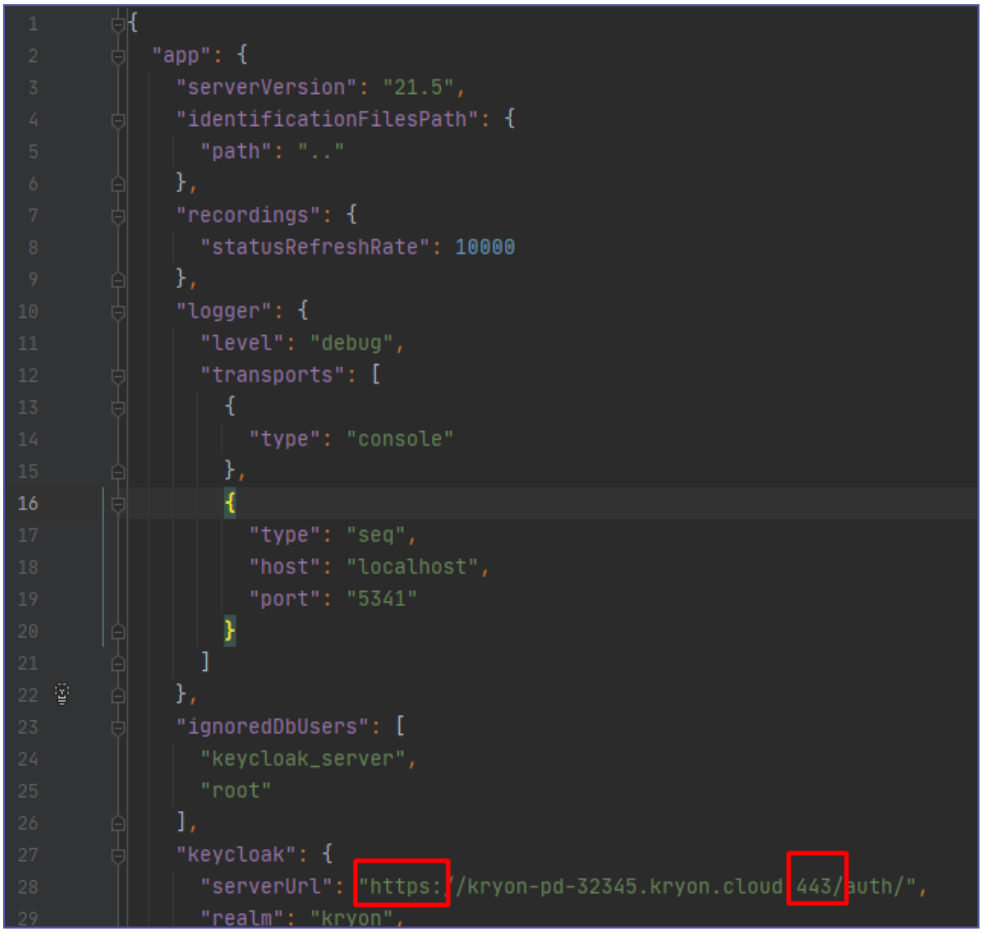
- "pdUrl": Change the URL protocol to **https**
- "pdPort": Change the port to **443**
- "pdAdminServices": Change the URL protocol to **https** and the port to **443**

EXAMPLE:

d. Verify the Process Discovery Admin configuration by opening the file `C:\Kryon\PDServer\Orchestrator\config\`**`production.json`** and verifying the following properties values:

- In the "keycloak" section, "serverUrl" property value is **https** and the port is **443**.

  EXAMPLE:



Step 4 - Start Process Discovery services

Start the following Kryon windows services:

- Kryon Server - Authentication Gateway
- Kryon Server - Authentication Server
- Kryon Server - Process Discovery Service

Step 5 -Verify Server TLS

Verify TLS by opening your browser and accessing the Process Discovery components by using **https://**:

- PD Console - `https://${SERVER_FQDN}/console`
- Seq logging server - `https://${SERVER_FQDN}/seq`

- Aerobase management - `https://${SERVER_FQDN}/auth/admin/kryon/console/#/realms/kryon`

Step 6 - Configure Discovery Robots

a. On your client machine, open

`%localappdata%l\Kryon\ActionsRecorder\config\`
`pddr.appsettings.config`

b. Modify the following parameters:

```
<add key="messagesProtocol" value="https"/> <!-- HTTPS protocol -->
<add key="messagesBrokerHost" value="${SERVER_FQDN}"/> <!-- with no protocol
prefix -->
<add key="messagesBrokerPort" value="443"/> <!-- HTTPS port -->
<add key="IDPuthUrl" value="https://${SERVER_FQDN}/auth/"/> <!-- with no port
specification -->
```

c. Re-run the Discovery Robot

Troubleshooting HTTPS/TLS Issues

Console is unavailable “ERR_SSL_PROTOCOL_ERROR”

1. Make sure your windows firewall doesn't block ports 80 (regular port) + 443 (secured port).
2. Open NGNIX logs to investigate the issue "C:\Kryon\IDP\Aerobase\Logs\nginx\error.log".

PDDR cannot connect to the server

1. Make sure your windows firewall doesn't block ports 80 (regular port) + 443 (secured port).
2. Open Wireshark on the server-side to see if the "client hello" message of the TLS handshake has arrived. If not, open Wireshark on the client-side and see if the "client hello" message has been sent.
3. Try to run PD's client on the same machine as the server.

APPENDIX C: Technical Data

The following sections include additional technical data.

In this Chapter:

HTTP Troubleshooting

Topics Overview:

PDDR is not recording

Open `%localappdata%\Kryon\ActionsRecorder\logs\pddr-Error.log` if exists to check if there are any communication errors.

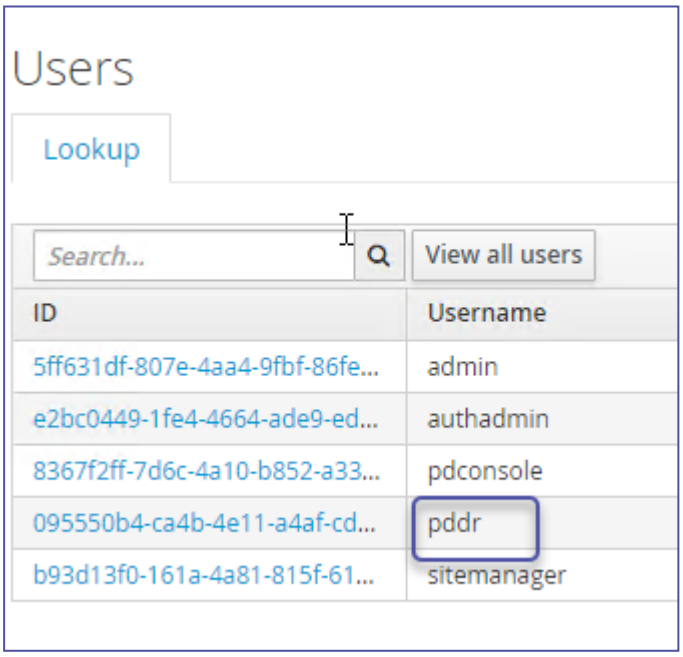
PDDR fails to connect - "Unable to resolve token Unauthorized"

When PDDR fails to connect, you might observe the following error on **pddr-error.log**:


```
{"unit":"pddr","uniqueIdentifier":"PDDR-
F874BEE1EE9772CBA8FF06066510BB0CE9381719826A7A3C48E0FD11E924CB28","level":"Error","ti
me":1618904387,"message":"Failed while sending a message. Error -> System.Exception:
Error on token request  ---> System.Exception: unable to resolve token
Unauthorized\r\n
at Kryon.ABPD.RecorderCommunication.GraphQL.GraphQlHttpMessageHandler.<TokenRequest>d_
_9.MoveNext()\r\n
--- End of stack trace from previous location where exception was thrown ---\r\n
at System.Runtime.ExceptionServices.ExceptionDispatchInfo.Throw()\r\n
at System.Runtime.CompilerServices.TaskAwaiter.HandleNonSuccessAndDebuggerNotification
(Task task)\r\n
at Kryon.ABPD.RecorderCommunication.GraphQL.GraphQlHttpMessageHandler.<GetToken>d__
10.MoveNext()\r\n
--- End of stack trace from previous location where exception was thrown ---\r\n
at System.Runtime.ExceptionServices.ExceptionDispatchInfo.Throw()\r\n
at System.Runtime.CompilerServices.TaskAwaiter.HandleNonSuccessAndDebuggerNotification
(Task task)\r\n
at Kryon.ABPD.RecorderCommunication.GraphQL.GraphQlHttpMessageHandler.<SendAsync>d__
7.MoveNext()\r\n
--- End of inner exception stack trace ---\r\n
at Kryon.ABPD.RecorderCommunication.GraphQL.GraphQlHttpMessageHandler.<SendAsync>d__
7.MoveNext()\r\n
--- End of stack trace from previous location where exception was thrown ---\r\n
at System.Runtime.ExceptionServices.ExceptionDispatchInfo.Throw()\r\n
at System.Runtime.CompilerServices.TaskAwaiter.HandleNonSuccessAndDebuggerNotification
(Task task)\r\n
at System.Net.Http.HttpClient.<FinishSendAsyncUnbuffered>d__59.MoveNext()\r\n
--- End of stack trace from previous location where exception was thrown ---\r\n
at System.Runtime.ExceptionServices.ExceptionDispatchInfo.Throw()\r\n
at System.Runtime.CompilerServices.TaskAwaiter.HandleNonSuccessAndDebuggerNotification
(Task task)\r\n
```

```
at GraphQL.Client.Http.GraphQLHttpClient.<SendHttpRequestAsync>d__28`1.MoveNext()\r\n
--- End of stack trace from previous location where exception was thrown ---\r\n
at System.Runtime.ExceptionServices.ExceptionDispatchInfo.Throw()\r\n
at System.Runtime.CompilerServices.TaskAwaiter.HandleNonSuccessAndDebuggerNotification
(Task task)\r\n
at GraphQL.Client.Http.GraphQLHttpClient.<SendQueryAsync>d__23`1.MoveNext()\r\n
--- End of stack trace from previous location where exception was thrown ---\r\n
at System.Runtime.ExceptionServices.ExceptionDispatchInfo.Throw()\r\n
at System.Runtime.CompilerServices.TaskAwaiter.HandleNonSuccessAndDebuggerNotification
(Task task)\r\n
at Kryon.ABPD.RecorderCommunication.HttpCommunicationManager.<SendRobotStatus>d__
29.MoveNext()\r\n
--- End of stack trace from previous location where exception was thrown ---\r\n
at System.Runtime.ExceptionServices.ExceptionDispatchInfo.Throw()\r\n
at System.Runtime.CompilerServices.TaskAwaiter.HandleNonSuccessAndDebuggerNotification
(Task task)\r\n
at Kryon.ABPD.RecorderCommunication.HttpCommunicationManager.<SendStatusUpdate>d__
24.MoveNext()"}
```

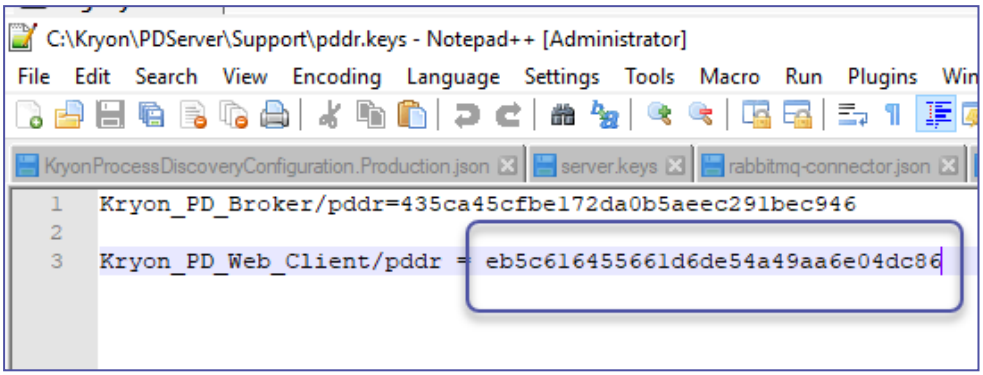
1. Make sure PDDR's Aerobase user is defined on the server:
   a. Open a browser on the server and navigate to `http://localhost/auth/admin/kryon/console/#/realms/kryon` (default credentials: authadmin/Kryon123!)
   b. On the left menu select **Users** > **View all users**
   c. Search for the "pddr" username and make sure it is defined (i.e., it exists in the table).



   PDDR user isn't define? Define it.

2. Make sure the password is correct:

a. Open `${InstallationFolder}\Kryon\PDServer\Support\pddr.keys` and verify that it contains the role **Kryon_PD_Web_Client/pddr** with an encrypted password.



b. Copy the keys file to PDDR's installation folder and restart PDDR.

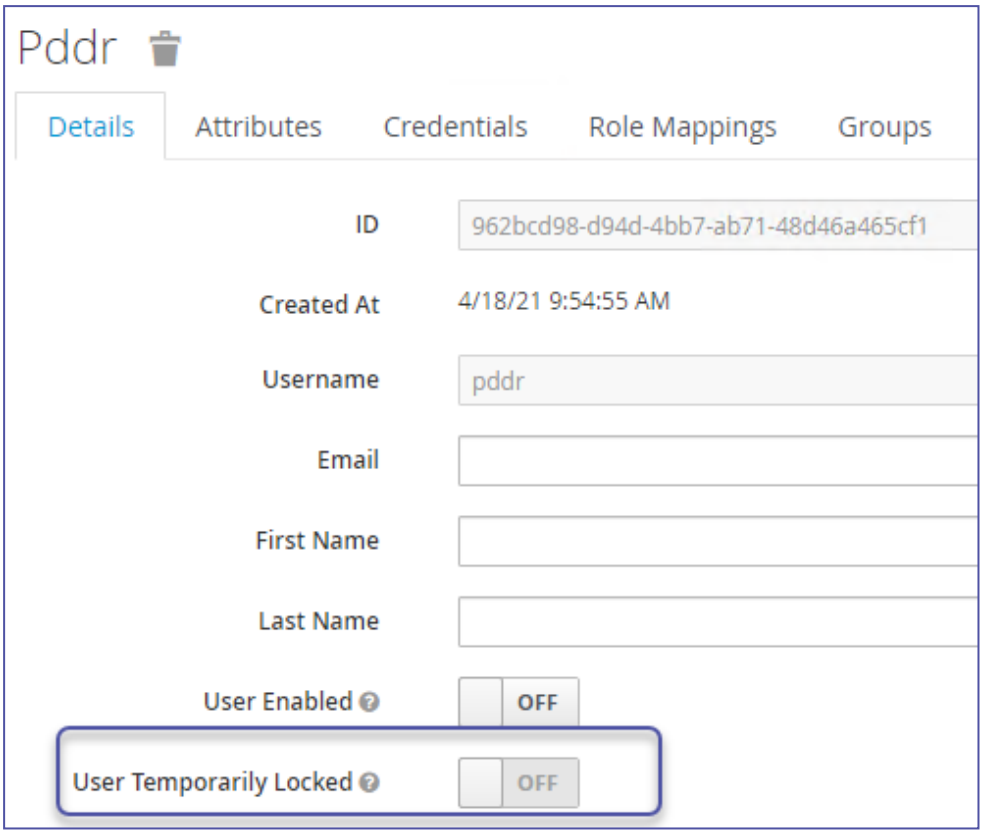
3. Make sure the PDDR user isn't locked on Aerobase:

   Aerobase might lock the pddr user after several failed connection attempts (incorrect credentials).

   To check if the PDDR user is locked:

   i. Navigate to Aerobase UI (`http://localhost/auth/admin/kryon/console/#/realms/kryon` default credentials: authadmin/Kryon123!) > **Users** > **PDDR** user > Click on the PDDR's ID > Details tab.

ii. Check if the user locked or not - unclock if locked:



PDDR Aerobase user is not defined

1. Open CMD as admin and run

```
cd C:\Kryon\PDServer\Support\kryon-admin-cli
set CONFIG_DIR=C:\Kryon\config
set NODE_ENV=prod
set KRYON_ENC_CFG=C:\Kryon\config\prod\general\kryon-decrypt.json
set NODE_PATH=C:\Kryon\PDServer\MicroServices\node_modules
node bin\cli.js --provider=aerobase --command=addUser --authUserName=authAdmin -
-authUserPassword=Kryon123! --userName=pddr --roleName=pd-robot --
vaultPath=C:\Kryon\PdServer\support\pddr.keys
```

2. Copy the keys file to PDDR's installation folder and restart PDDR

I need the Aerobase password

1. Set the password when adding the user (instead of generating one):
   a. Open CMD as admin and run

```
cd C:\Kryon\PDServer\Support\kryon-admin-cli
set CONFIG_DIR=C:\Kryon\config
set NODE_ENV=prod
set KRYON_ENC_CFG=C:\Kryon\config\prod\general\kryon-decrypt.json
set NODE_PATH=C:\Kryon\PDServer\MicroServices\node_modules
node bin\cli.js --provider=aerobase --command=addUser --
authUserName=authAdmin --authUserPassword=Kryon123! --userName=pddr --
auth=Pd123456789!! --roleName=pd-robot --
vaultPath=C:\Kryon\PdServer\support\pddr.keys
```

   b. Copy the keys file to PDDR's installation folder and restart PDDR
2. Reset the password:
   a. Open CMD as admin and run

```
cd C:\Kryon\PDServer\Support\kryon-admin-cli
set CONFIG_DIR=C:\Kryon\config
set NODE_ENV=prod
set KRYON_ENC_CFG=C:\Kryon\config\prod\general\kryon-decrypt.json
set NODE_PATH=C:\Kryon\PDServer\MicroServices\node_modules
node bin\cli.js --provider=aerobase --command=changeUserPassword --
authUserName=authAdmin --authUserPassword=Kryon123! --userName=pddr --
auth=Kryon123! --vaultPath=C:\Kryon\PdServer\support\pddr.keys
```

   b. Copy the keys file to PDDR's installation folder and restart PDDR
3. Decrypt the password:
   a. Open PDDR's keys file
   `${InstallationFolder}\Kryon\PDServer\Support\`**`pddr.keys`**
   b. Copy the encrypted password from line:
   `Kryon_PD_Web_Client/pddr=${TheEncryptedPasswordYouShouldCopy}`
   c. Open CMD as admin and run

```
cd C:\Kryon\PDServer\Support\kryon-admin-cli
set CONFIG_DIR=C:\Kryon\config
set NODE_ENV=prod
set KRYON_ENC_CFG=C:\Kryon\config\prod\general\kryon-decrypt.json
set NODE_PATH=C:\Kryon\PDServer\MicroServices\node_modules
node bin\cli.js --provider=vault --command=decryptPhrase --
phrase=${TheEncryptedPasswordYouShouldCopy}
```

PDDR fails to connect - "connected host has failed to respond"

When PDDR fails to connect, you might observe the following error on **pddr-error.log**:

```
unable to connect to the remote server --->
System.Net.Sockets.SocketException: A connection attempt failed because the connected
party did not properly respond after a period of time,
or established connection failed because connected host has failed to respond
10.225.5.4:80
```

1. Open port 80 on the server:
   a. Click **Start** > **Windows Defender Firewall** > **Advanced Settings**
   b. On the left menu, select **Inbound Rules** > **New Rule**
   c. Select the tab **Protocols and Ports**
   d. For Local port, set **Specific Ports: 80** .
   e. In the prompted window, allow the connections (default) > select all (default) > give the rule a meaningful name > Finish
   f. Save / OK.

PDDR fails to connect - "The remote name could not be resolved"

When PDDR fails to connect, you might observe the following error on **pddr-error.log**:

```
System.Exception: Error on token request  ---> System.Net.Http.HttpRequestException:
An error occurred while sending the request. ---> System.Net.WebException: The remote
name could not be resolved: 'kryon-pd-22066.kryon.cloud'\r\n
at System.Net.HttpWebRequest.EndGetRequestStream(IAsyncResult asyncResult,
TransportContext& context)\r\n
at System.Net.Http.HttpClientHandler.GetRequestStreamCallback(IAsyncResult ar)\r\n
— End of inner exception stack trace ---\r\n
at System.Runtime.ExceptionServices.ExceptionDispatchInfo.Throw()\r\n
at System.Runtime.CompilerServices.TaskAwaiter.HandleNonSuccessAndDebuggerNotification
(Task task)\r\n
at System.Net.Http.HttpClient.<FinishSendAsyncBuffered>d_58.MoveNext()\r\n
--- End of stack trace from previous location where exception was thrown ---\r\n
at System.Runtime.ExceptionServices.ExceptionDispatchInfo.Throw()\r\n
at System.Runtime.CompilerServices.TaskAwaiter.HandleNonSuccessAndDebuggerNotification
(Task task)\r\n
at System.Runtime.CompilerServices.TaskAwaiter`1.GetResult()\r\n
at
Kryon.ABPD.RecorderCommunication.GraphQL.GraphQlHttpMessageHandler.<TokenRequest>d9.Mo
veNext() in C:\\git\\Kryon_
PD\\Src\\ABPD\\DiscoveryRobot\\Kryon.ABPD.RecorderCommunication\\GraphQLGraphQLHttpMes
sageHandler.cs:line 80\r\n
--- End of stack trace from previous location where exception was thrown ---\r\n
at System.Runtime.ExceptionServices.ExceptionDispatchInfo.Throw()\r\n
at System.Runtime.CompilerServices.TaskAwaiter.HandleNonSuccessAndDebuggerNotification
(Task task)\r\n
at System.Runtime.CompilerServices.TaskAwaiter`1.GetResult()\r\n
at
Kryon.ABPD.RecorderCommunication.GraphQL.GraphQlHttpMessageHandler.<GetToken>d10.MoveN
ext() in C:\\git\\Kryon_
```

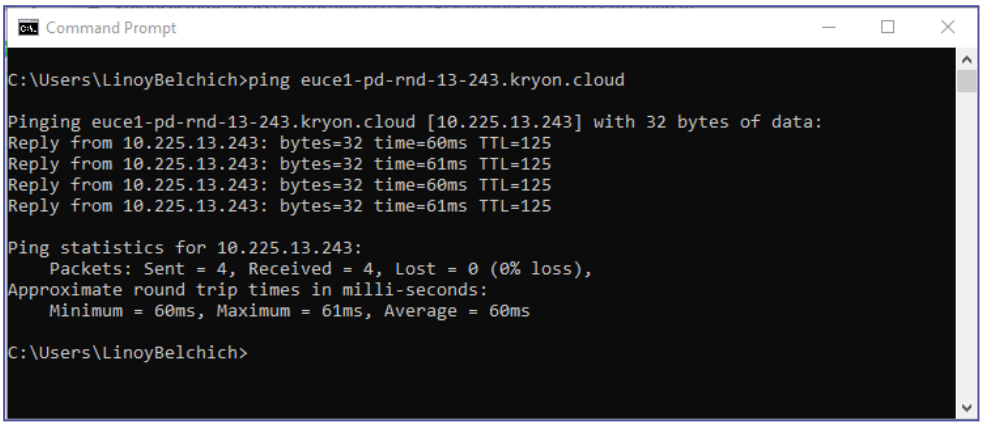
```
PD\\Src\\ABPD\\DiscoveryRobot\\Kryon.ABPD.RecorderCommunication\\GraphQLGraphQLHttpMes
sageHandler.cs:line 96\r\n
--- End of stack trace from previous location where exception was thrown ---\r\n
at System.Runtime.ExceptionServices.ExceptionDispatchInfo.Throw()\r\n
at System.Runtime.CompilerServices.TaskAwaiter.HandleNonSuccessAndDebuggerNotification
(Task task)\r\n
at System.Runtime.CompilerServices.TaskAwaiter`1.GetResult()\r\n
at
Kryon.ABPD.RecorderCommunication.GraphQL.GraphQlHttpMessageHandler.<SendAsync>d7.MoveN
ext() in C:\\git\\Kryon_
PD\\Src\\ABPD\\DiscoveryRobot\\Kryon.ABPD.RecorderCommunication\\GraphQLGraphQLHttpMes
sageHandler.cs:line 49\r\n
--- End of inner exception stack trace ---\r\n
at
Kryon.ABPD.RecorderCommunication.GraphQL.GraphQlHttpMessageHandler.<SendAsync>d7.MoveN
ext() in C:\\git\\Kryon_
PD\\Src\\ABPD\\DiscoveryRobot\\Kryon.ABPD.RecorderCommunication\\GraphQLGraphQLHttpMes
sageHandler.cs:line 53\r\n
--- End of stack trace from previous location where exception was thrown ---\r\n
at System.Runtime.ExceptionServices.ExceptionDispatchInfo.Throw()\r\n
at System.Runtime.CompilerServices.TaskAwaiter.HandleNonSuccessAndDebuggerNotification
(Task task)\r\n
at System.Net.Http.HttpClient.<FinishSendAsyncUnbuffered>d59.MoveNext()\r\n
--- End of stack trace from previous location where exception was thrown ---\r\n
at System.Runtime.ExceptionServices.ExceptionDispatchInfo.Throw()\r\n
at System.Runtime.CompilerServices.TaskAwaiter.HandleNonSuccessAndDebuggerNotification
(Task task)\r\n
at GraphQL.Client.Http.GraphQLHttpClient.<SendHttpRequestAsync>d28`1.MoveNext()\r\n
--- End of stack trace from previous location where exception was thrown ---\r\n
at System.Runtime.ExceptionServices.ExceptionDispatchInfo.Throw()\r\n
at System.Runtime.CompilerServices.TaskAwaiter.HandleNonSuccessAndDebuggerNotification
(Task task)\r\n
at GraphQL.Client.Http.GraphQLHttpClient.<SendQueryAsync>d23`1.MoveNext()\r\n
--- End of stack trace from previous location where exception was thrown ---\r\n
at System.Runtime.ExceptionServices.ExceptionDispatchInfo.Throw()\r\n   at
System.Runtime.CompilerServices.TaskAwaiter.HandleNonSuccessAndDebuggerNotification
(Task task)\r\n
at System.Runtime.CompilerServices.TaskAwaiter`1.GetResult()\r\n
at
Kryon.ABPD.RecorderCommunication.HttpCommunicationManager.<SendRobotStatus>d29.MoveNex
t() in C:\\git\\Kryon_
PD\\Src\\ABPD\\DiscoveryRobot\\Kryon.ABPD.RecorderCommunicationHttpCommunicationManage
r.cs:line 153\r\n
--- End of stack trace from previous location where exception was thrown ---\r\n
at System.Runtime.ExceptionServices.ExceptionDispatchInfo.Throw()\r\n
at System.Runtime.CompilerServices.TaskAwaiter.HandleNonSuccessAndDebuggerNotification
(Task task)\r\n
at System.Runtime.CompilerServices.TaskAwaiter`1.GetResult()\r\n
at Kryon.ABPD.RecorderCommunication.HttpCommunicationManager.<SendStatusUpdate>d_
24.MoveNext() in C:\\git\\Kryon_
PD\\Src\\ABPD\\DiscoveryRobot\\Kryon.ABPD.RecorderCommunicationHttpCommunicationManage
r.cs:line 63"}
```


1. Fix the DNS problem by updating the hosts file on the client machine (PDDR machine) by:
   a. Open `C:\Windows\System32\drivers\etc\hosts` file in notepad editor
   b. Add the server IP + FQDN as following:

```
10.225.13.243 euce1-pd-rnd-13-243.kryon.cloud
```

c. Save the file and re-run PDDR

d. You can verify the fix by using the ping command to the machine FQDN.
EXAMPLE:



PDDR fails to connect - "The HTTP request failed with status code NotFound"

When PDDR fails to connect, you might observe the following error on **pddr-error.log**:

```
{"unit":"pddr","uniqueIdentifier":"PDDR-
5EA42760CF6E32E96DD266CAE2F9827D16E8C1FAB3BB01A859F1DA07F59D12C8","level":"Error","tim
e":1618830874,"message":"Failed while sending a message. Error ->
GraphQL.Client.Http.GraphQLHttpRequestException: The HTTP request failed with status
code NotFound\r\n
at GraphQL.Client.Http.GraphQLHttpClient.<SendHttpRequestAsync>d__28`1.MoveNext()\r\n
--- End of stack trace from previous location where exception was thrown ---\r\n
at System.Runtime.ExceptionServices.ExceptionDispatchInfo.Throw()\r\n
at System.Runtime.CompilerServices.TaskAwaiter.HandleNonSuccessAndDebuggerNotification
(Task task)\r\n
at GraphQL.Client.Http.GraphQLHttpClient.<SendQueryAsync>d__23`1.MoveNext()\r\n
--- End of stack trace from previous location where exception was thrown ---\r\n
at System.Runtime.ExceptionServices.ExceptionDispatchInfo.Throw()\r\n
at System.Runtime.CompilerServices.TaskAwaiter.HandleNonSuccessAndDebuggerNotification
(Task task)\r\n
at System.Runtime.CompilerServices.TaskAwaiter`1.GetResult()\r\n
at Kryon.ABPD.RecorderCommunication.HttpCommunicationManager.<SendRobotStatus>d__
32.MoveNext() in C:\\git\\Kryon_
PD\\Src\\ABPD\\DiscoveryRobot\\Kryon.ABPD.RecorderCommunication\\HttpCommunicationMana
ger.cs:line 155\r\n
--- End of stack trace from previous location where exception was thrown ---\r\n
at System.Runtime.ExceptionServices.ExceptionDispatchInfo.Throw()\r\n
at System.Runtime.CompilerServices.TaskAwaiter.HandleNonSuccessAndDebuggerNotification
(Task task)\r\n
at System.Runtime.CompilerServices.TaskAwaiter`1.GetResult()\r\n
at Kryon.ABPD.RecorderCommunication.HttpCommunicationManager.<SendStatusUpdate>d__
27.MoveNext() in C:\\git\\Kryon_
PD\\Src\\ABPD\\DiscoveryRobot\\Kryon.ABPD.RecorderCommunication\\HttpCommunicationMana
ger.cs:line 64"}
```

This is a proxy problem - you should debug it on the server machine by checking NGINX files.

EXAMPLE:

`C:\Kryon\IDP\Aerobase\Logs\nginx\`**`access.log`** (log files)

`C:\Kryon\IDP\Aerobase\Data\nginx\conf.d\`**`kryon-pddr-locations.import`** (config file)

Discovery Robots and Application Monitoring

Discovery Robots run silently on your employees' machines, collecting data on how they utilize business applications to perform their daily tasks. On the Applications tab of the Process Discovery Console, you can define which applications the Discovery Robots should monitor and which to ignore. You can choose either one of three approaches:

- **Option 1**: Define the list of the only applications you want the Discovery Robot to monitor **(Recommended)**
- **Option 2**: Define a list of application to exclude
- **Option 3**: Set the robot to record and monitor all applications

BEST PRACTICE
Applications to include:

What is the best approach to defining the applications to include in the recorded sessions?

Day 1 of recordings: Set applications to exclude. Application that you know you don't want . At the end of the day, review the day's output.

Day 2 of recordings: Set the applications to include based on the first day's results.

Regardless of the approach you choose, there are two kinds of application types you can monitor:

- **Desktop applications** – for example, Outlook. Included in this category are *terminal emulators*, applications that allow remote access to a legacy text terminal
- **Web applications** – a URL like video.google.co.uk

NOTE
Recording keystrokes from different languages

Discovery Robots can record keystrokes from a number of keyboard languages, including:

Albanian, Latvian, Armenian, Lithuanian, Bulgarian, Macedonian, Catalan, Norwegian, Croatian, English, Polish, Czech, Portuguese, Danish, Romanian, Dutch, Russian, Estonian, Slovak, Finnish, Slovene, French, , Spanish, German, Swedish, Greek (Modern), Turkish,

Hungarian, Ukrainian, Icelandic, Russian, Italian, Hebrew

Remote Desktop Workstations

When the discovery robots are installed on RDP client workstations, application filtering is not done on recordings from the remote computer.

Terminal emulators

Supported character-sets include: Latin, Cyrillic, and Hebrew.

Supported screen backgrounds: any color but white

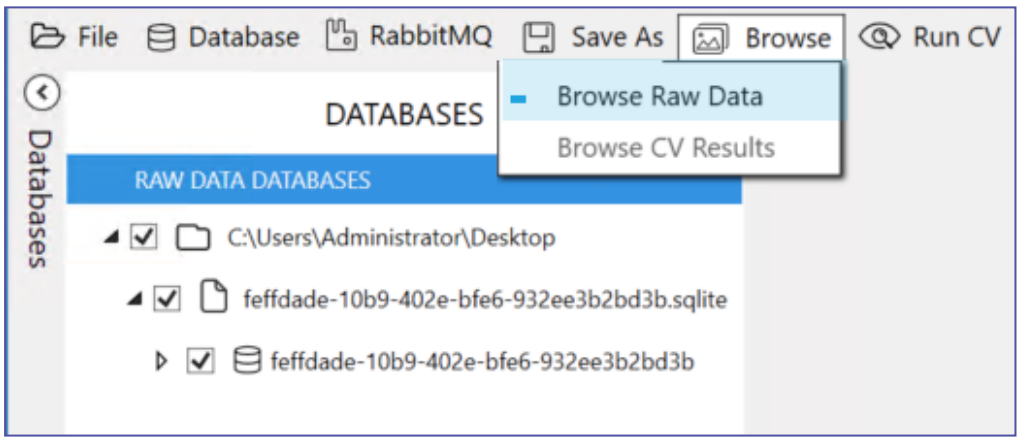
- I got here while configuring, take me back to Defining Applications to Record -

Browsing Recorded Sessions Data

Another useful feature of the Toolkit is that you can use it to browse the raw data of the Recorded Sessions , review the full recordings or monitor the PD process. The Browse feature allows you to view all data captured during the session, from beginning to end, both images and metadata.

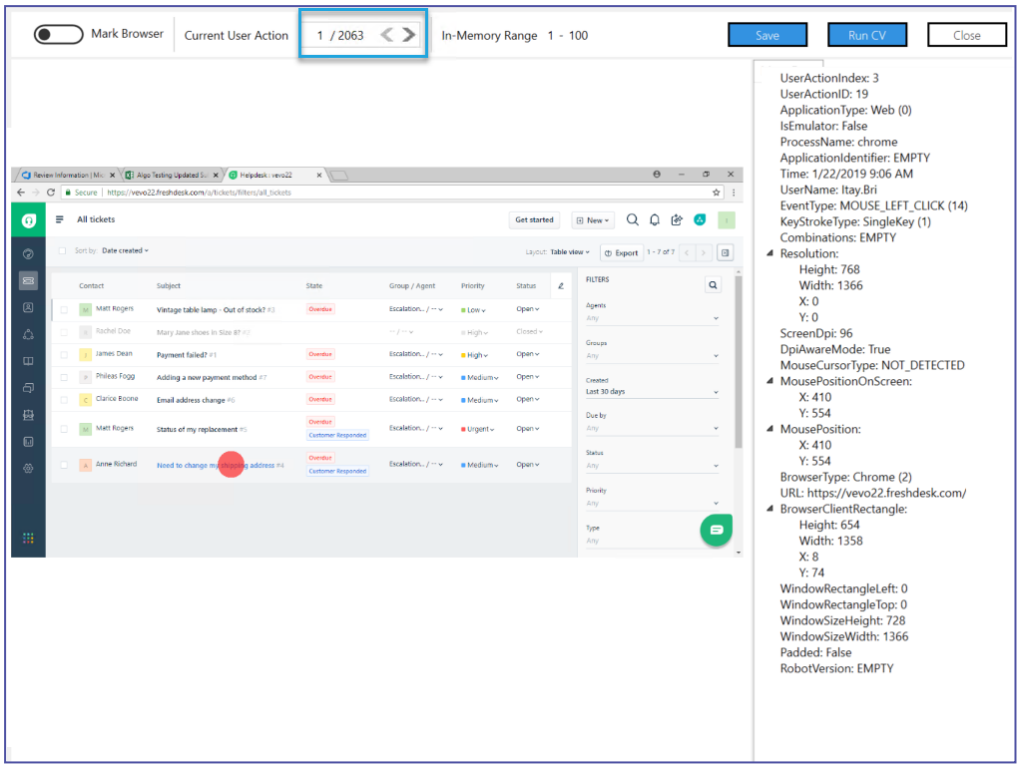
Browsing recorded sessions:

1. Open the Kryon Process Discovery Toolkit, **Toolkit.exe**, located in `{InstallFolder}\PDServer\Support\Toolkit`.
2. Download all or only selected recording as explained in Exporting Recorded Sessions.
3. You can only check one recording at a time - make sure only one item is checked in the Databases list on the right pane. From the menu, select **Browse> Browse Raw Data**.



4. You can scroll through the images by clicking the **<** and **>** buttons or using the

keyboard right or left arrow.



Installed Components & Software

Discovery server

The following software is automatically installed on the **Discovery Server** by the Kryon Process Discovery server installation package, if not previously installed:

- Microsoft .NET Framework 4.7.2
- Microsoft .NET Core 3.1.10 – Windows Server Hosting
- Microsoft Visual C++ 2015-2019 Redistributable (x64)
- RabbitMQ Server (used only for internal server communication)
- Erlang OTP (the programming language on which RabbitMQ is built)
- NodeJS (JavaScript runtime used by Kryon Process Discovery Admin)
- Seq (centralized logging component)
- Tesseract in case masking is installed

The following software can be optionally installed by the Kryon Process Discovery server installation package:

- HeidiSQL (database viewer)
- Notepad++

Discovery robot

The following software is automatically installed on the client machine during the **Discovery Robot** installation, if not previously installed:

- Microsoft .NET Framework 4.7.2
- Microsoft Visual C++ 2015-2019 Redistributable (x64/x86 as appropriate)

Third party components

Third party software provided as part of or with the Licensed Product is solely governed by its respective license terms as set forth in:

https://public.kryon.io/#PD-Versions/21.3/Documents/

Discovery Server Installation Configuration File

Configuration of the **Discovery Server** installation is done by editing the JSON file provided with the installation package, `PDServer64BitSetup.exe.json`.

To edit the JSON file:

1. Open the JSON file with a text editor
2. Enter the value for the relevant parameter between the corresponding double-quotes, for example:

   `"TENANTS_DB_USER_PASSWORD": "pdpassword1234!",`

   Note: The syntax for specifying folder and file locations in JSON uses a double backslash in each location in which Windows syntax would use a single backslash, for example:

   `C:\\Program Files\\MariaDB\\`
3. Save the JSON file

The following is a full explanation of **Discovery Server** JSON configuration file parameters:

Parameter Name	Description
`INSTALLFOLDER`	Folder in which the core **Discovery Server** files will be installed The format for the location of the (missing or bad snippet) install folder is `{local drive}:\{InstallFolder}`. • The **InstallFolder** should be targeted to a local folder on a drive with more than 500GB of free disk space The `InstallFolder` can't be the root folder of a drive (for example, `C:\` is not allowed) The `{InstallFolder}` folder name can't include spaces (for example, `.C:\Install Folder` is not allowed The `InstallFolder` folder path cannot be

Parameter Name	Description
	longer than 19 characters (20 if a backslash is added at the end) **Default value:** `{Local fixed drive with the most free space}:\Kryon`
ENABLE_TLS_IN_REGISTRY	If set to `True`, installation will modify the registry of the machine on which the Discovery Server is installed to: • enable SSL/TLS 1.2; and • disable earlier versions of SSL/TLS and SSL Must be set to `True` if you will be deploying SSL/TLS Configuration **Default value:** `True`
LOCAL_MARIADB_USE	When set to `True`: • If there is no preexisting instance of MariaDB installed on the local machine, the Kryon Process Discovery Server installation package will automatically install MariaDB • If there is a preexisting instance of MariaDB on the local machine (and the installation package is able to connect to it using the values specified in the DB_ROOT_NAME, DB_ROOT_PASSWORD, DB_PORT, and DB_SERVICE_NAME parameters), the Kryon Process Discovery Server installation package will create Kryon Process Discovery application database #1 and Kryon Process Discovery application database #2 within it **Default value:** `True`
DB_INSTALLDIR	The database install directory **Default value:** `C:\Kryon\MariaDB`
DB_SERVICE_NAME	Used when **LOCAL_MARIA_DB_USE** is set to `True` The name of the MariaDB Windows service • If there is no preexisting instance of MariaDB, the

Parameter Name	Description
	installation package will configure MariaDB to use this service name • If there is a preexisting instance of MariaDB, the installation package will attempt to connect to it using this service name (should match the service name configured for this instance) **Default value:** `MariaDB`
DB_SERVER	Discovery Server IP address or FQDN (Fully Qualified Domain Name) of the machine on which the database is installed • Do not change the default value if the database is installed on the same machine as the **Discovery Server** • Must be changed from the default value if the database is installed on a remote machine **Default value:** `<local machine name>`
DB_PORT	The network port by which the **Discovery Server** will access the database **Default value:** `3306`
DB_ROOT_NAME	The user that accesses the database as a superuser (e.g., for creating database schema, etc.) **Default value:** `root`
DB_ROOT_PASSWORD	The password used to access the database as a superuser (e.g., for creating database schema, etc.) • Must match the root password created during database instance installation **Default value:** `Kryon2020!`
DB_USER_NAME	The user of **Discovery Server** that accesses the database as a regular user **Default value:** `pdbdev`

Parameter Name	Description
DB_USER_PASSWORD	The password used by the **Discovery Server** to access the database as a regular user **Default value:** `Kryon2020!`
PD_DEFAULT_TENANT_ID	ID of the default Kryon Process Discovery Team • If you wish to change the default Team ID, you must do so here (i.e., it cannot be changed post-installation) • You can add additional Teams here by separating the Team IDs with commas (e.g.: `default,tenant2,tenant3`) • **Team ID**: can be 3 to 42 characters - use only **lower-case letters** and numbers (no spaces or special characters allowed) • Additional Teams can also be added at any time after installation. For additional information, see Managing Teams. **Default value:** `default`
LOCAL_RABBITMQ_USE	When set to `True`: • If there is no preexisting instance of RABBITMQ installed on the local machine, the Kryon Process Discovery Server installation package will automatically it • If there is a preexisting instance of RABBITMQ on the local machine, the Kryon Process Discovery Server installation package will create the Process Library database and authentication platform database schema within it **Default value:** `True`
MESSAGES_BROKER_HOST	The FQDN (Fully Qualified Domain Name, i.e., DNS name) of the machine on which RabbitMQ Server will be installed • **Recommended:** Leave empty so that the

Parameter Name	Description
	package will be installed on the local machine **Default value:** `localhost`
MESSAGES_BROKER_PORT	The network port for communication between the **Discovery Robots** and RabbitMQ Server • **Recommended:** Leave the default value of `5672` when not using TLS, or change to `5671` when using TLS. This will minimize configuration changes required post-installation. **Default value:** `5672`
RABBITMQ_ADMIN_NAME	Used when **LOCAL_RABBITMQ_USE** is set to `True` • The installation package creates a regular user account **Default value:** `admin`
RABBITMQ_ADMIN_PASSWORD	The password used to access RabbitMQ Server **Default value:** `Kryon2020!`
DOC_MANAGER_PORT	The internal port for communication between the **Discovery Server** and the **Document Manager** **Default value:** `50007`
PDSERVER_PORT	The PD Admin (formerly *Orchestrator*) port **Default value:** `8788`
AB_APPLICATION_SERVER	The FQDN (Fully Qualified Domain Name, i.e., DNS name) of the User Management Tool platform application server • **Recommended:** Leave empty so that the application server will be set to local machine • If not left empty, should match the value of **CX_APPLICATION_SERVER** **Default value:**

Parameter Name	Description
AB_CONSOLE_PORT	**Recommended:** Do not change from default value **Default value:** `5058`
AB_KEYCLOAK_AUTHADMIN_PASSWORD	The default password for authadmin user of Keycloak **Default value:** `Kryon123!`
AB_SERVICE_LOGON_USER	The user name and password that will be used to access the authentication platform Windows service
AB_SERVICE_LOGON_PASSWORD	
CX_APPLICATION_SERVER	The FQDN (Fully Qualified Domain Name, i.e., DNS name) of the **Process Library** application server • **Recommended:** Leave empty so that the application server will be set to local machine • If not left empty, should match the value of **AB_APPLICATION_SERVER**
CX_PORT_AUTH	Network ports for used communication between Process Library and its backend APIs **Default value: CX_PORT_AUTH** `5002` **Default value: CX_PORT_GATEWAY** `5001` **Default value: CX_PORT_DISCOVERY** `5004` **Default value: CX_PORT_SETTINGS** `5003`
CX_PORT_GATEWAY	
CX_PORT_DISCOVERY	
CX_PORT_SETTINGS	
PORT_NGINX	**Default value: PORT_NGINX** `80`
INSTALL_NPP	if set to `True`, Notepad++ source code editor is installed • If Notepad++ already is installed on the server, a new instance is not installed **Default value:** `True`

Configuring Discovery Robots

The basic Discovery Robot setting are configured upon installation through the installation wizard. Once **Discovery Robots** are installed on the employee workstation, you can apply additional Discovery Robot configurations by editing the Discovery Robot Configuration File. You should receive from your organization the specific configuration requirements for each **Discovery Robot**.

Discovery Robot Configuration File

You can locate and edit the Discovery Robot settings in the Discovery Robot Configuration File.

The Discovery Robot Configuration file name and location depends on the following:

Before the first run of Process Discovery Robot:
`${InstallationFolder}/PDDR/pddr.appsettings_template.config`
(The usual path is: C:\Program Files\PDDR)

After the first run of Process Discovery Robot:
`%localappdata%/Kryon/ActionsRecorder/config/pddr.appsettings.config`

Parameter	Description/change
`<add key="messagesBrokerHost" value=" FQDN /IP Address"/>`	Sets the address of the **Discovery Server** `<add key="messagesBrokerHost" value="{{Discovery Server IP address or FQDN (Fully Qualified Domain Name)}}"/>` • **MUST** be changed from the default value of `localhost`
`<add key="messagesBrokerPort" value="port#"/>`	Sets the port number of the Process Discovery server. You can edit and change the port number. The default port number is `5672`.
`<add key="stealthMode" value="false"/>`	To hide **Discovery Robot's** tray icon on the machine's end-user(s), change the key value to `"true"`.
`<add key="persistRecords" value="false"/>`	By default, recordings by **Discovery Robots** are not maintained on the employee workstation once they have been uploaded to the **Discovery Server**.

Parameter	Description/change
	To maintain the recordings on the workstation after uploading to the **Discovery Server**, for debugging purposes, change the key value to "`true`".
`<add key="HashUserName" value="false"/>`	By default, the user name is visible in the **Discovery Robot's** data. To comply with privacy regulations (such as the GDPR) and/or company-issued privacy and security policies, you can hash the user name from all recorded/transmitted data by setting the option to hash (i.e., encrypt) the user name. To hash the username, change the key value to "`true`".
`<add key="standaloneMode" value="false"/>`	In standalone mode, the **Discovery Robot** records user actions while they remain disconnected from the **Discovery Server** • `false` = the **Discovery Robot** is not in standalone mode • `true` = the **Discovery Robot** is in standalone mode **Default Value** `false` For additional configuration of standalone robots, see the section **Standalone Discovery Robots** in *Process Discovery User Guide*.
`<add key="applicationsConfigPath" value="."/>`	Used when `standaloneMode` is set to `True` • Sets the path to the local Applications for Discovery configuration (`blwl.json`) file that defines which applications are recorded • "." indicates same folder Discovery Robot's exe file **Default Value** `"."`
`<add`	Used when `standaloneMode` is set to `True`

Parameter	Description/change
`key="offlineRecordsStoreLimit" value=2000/>`	• Defines the maximum number of records to save locally **Default Value** `2000`
`<add key="EmulatorCommandMaxLength" value="5"/>`	Due to PII and GDPR regulations, all input fields of emulator applications are automatically masked. You can set to unmask up to a configurable number of characters, so they can be interpreted as actions. The available range is 1 to 10. **Default Value** `5`
`<add key="CaptureScreenshots" value="true"/>`	By default, the parameter is set to capture screenshots when recording sessions (`"true"`). To record work-patterns and collect metadata without capturing screenshots, change the key value to `"false"`.
`<add key="UnicodeHookingProvider" value="false"/>`	Sets which hooking provider to use for recording keyboard strokes. Unicode capabilities enable support for multiple language keyboards. • `false` = Use a legacy provider that supports only ASCII • `true` = Use a provider that supports both ASCII and Unicode (for multiple language keyboards) **Default Value** `false`
• `<add key="TlsEnabled" value="false"/>` • `<add key="TlsServer" value=""/>` • `<add key="TlsPort" value="5671"/>`	To configure SSL/TLS on **Discovery Robots**, edit the keys as detailed below: • `y="TlsEnabled" value="true"/>` • `<add key="TlsServer" value="{certificate_server_name}"/>` ◦ `{certificate_server_name}` must be the name exactly as specified on the server certificate file.

Parameter	Description/change
	• `<add key="TlsPort" value="{ssl_listeners_port"/>` ◦ Change this key only if you are using a port other `5671` for SSL/TLS communications. ◦ This should be the same port as you set for `ssl_listeners` (in the `rabbitmq.config` file) when configuring the Discovery Server. **NOTE:** Only SSL/TLS v1.2 is supported.

`<add key="CaptureMethod" value="MIRROR_DRIVER"/>` *(default)*
 	
Other value option:
`"DESKTOP_DUPLICATION"`
`"GDI"`
	The 'CaptureMethod' parameters sets the PDDR capturing method. By default, the capturing method is set to "MIRROR_DRIVER". However, you might need to change the value to "DESKTOP_DUPLICATION" in the following specific case:
	When working via RDP (accessing a remote desktop from your local server) on a remote machine with PDDR and **Windows 10** installed.
	Once you change the key value and save your changes, make sure to restart the PD service.
	In case there is a failure in Mirror driver / desktop duplication - the fallback is GD.

Standalone Discovery Robots

A **Discovery Robot** in standalone mode works offline from the server. The major things to keep in mind if you want to use a standalone robot are:

- The **Discovery Robot** doesn't upload recordings to the server. They are stored on the user workstation.
- The default number of actions that are recorded and stored on the employee workstation is 2000 recorded actions. When the limit is reached, the **Discovery Robot** stops recording (You can configured the limit by editing the `offlineRecordsStoreLimit` parameter in the `pddr.appsettings.config`, explained below)
- Since the **Discovery Robot** isn't connected to the **Discovery Server**, it uses its own local configuration file (`blwl.JSON`) to define which applications it records
- You can find recorded data saved on the employee workstation in `%Localappdata%\Kryon\ActionsRecorder\db`

Steps for setting up a standalone Robot

1. Open the file `%LOCALAPPDATA%\Kryon\ActionsRecorder\config/pddr.appsettings.config` with a text editor and edit the keys as explained:
   a. In `pddr.appsettings.config`, change the value of the following keys, and save:

      change:

      ```
      <add key="standaloneMode" value="false"/>
      ```

      to:

      ```
      <add key="standaloneMode" value="true"/>
      ```

   b. The `offlineRecordsStoreLimit` parameter defines the number of actions that are recorded and stored on the local device. The default is 2000 recorded actions.

      To change the recording limit:

      In `pddr.appsettings.config`, change the value of the following keys, and save:

      change:

      ```
      <add key="offlineRecordsStoreLimit" value="2000"/>
      ```

to:

```
<add key="offlineRecordsStoreLimit" value="[new value]"/>
```

2. Create and download a copy of the configuration file (`blwl.JSON`) from the **Discovery Server**:

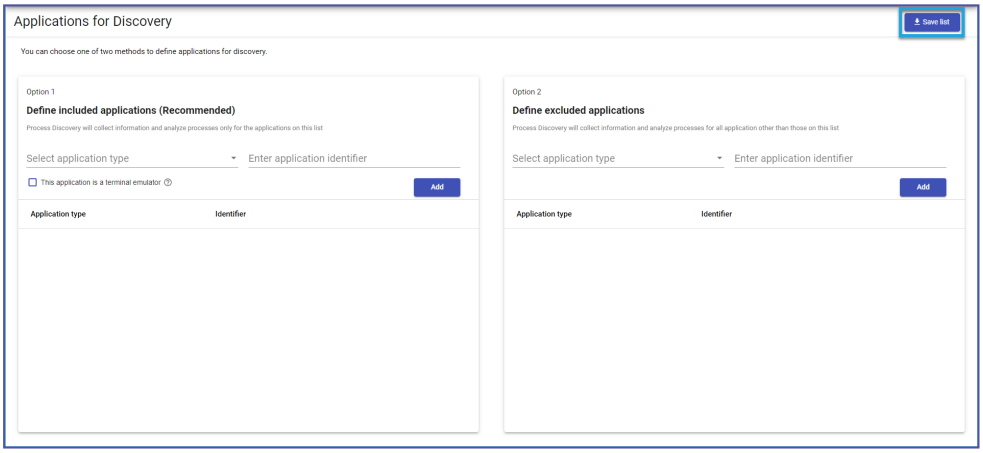
   a. Access Kryon Process Discovery Console > **Settings** page

   b. Click the **Applications** tab

      The Applications for Discovery screen opens

   c. Define applications for discovery in the list of either option 1 or 2 (we recommend option 1), as explained in Discovery Robots and Application Monitoring

   d. Click the **Save List** button to download the `blwl.JSON` file



   e. Save `blwl.JSON` in the folder location as defined in the `applicationsConfigPath` parameter in `pddr.appsettings.config`. The default location is the same folder as `pddr.appsettings.config`.

      To change the path:

      In `pddr.appsettings.config`, change the value of the following keys, and save:

      change:

      ```
      <add key="applicationsConfigPath" value="."/>
      ```

      to:

      ```
      <add key="applicationsConfigPath" value="[new path]"/>
      ```

      Note: The syntax for specifying folder and file locations in JSON uses a double backslash in each location in which Windows syntax would use a single

backslash, for example:

```
C:\\Program Files\\YourFolder\\
```

Image Masking

Hide potentially sensitive recorded information collected by Process Discovery.

When masking is enabled, every screen capture displayed on the Console goes through the masking mechanism that:

1. Analyzes all the text on the recorded screen
2. Identifies potentially sensitive information
3. Blurs out the target text

Enabling Image-Masking

You can enable image-masking before installing Process Discovery v20.11 (recommended), or after installation.

Then, you can also set he Image Masking Scheduler

Option 1: Pre-Installation

Before you run the Process Discovery installation, perform the following:

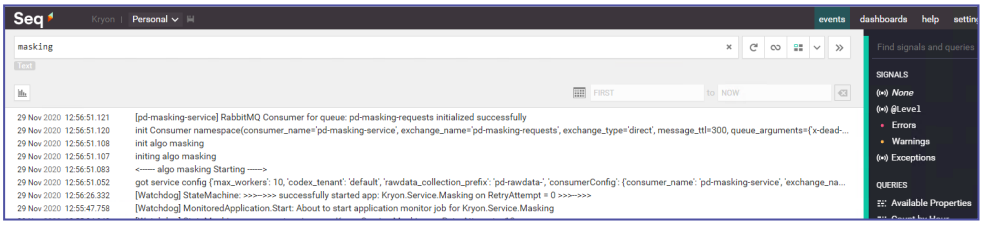
1. Open the `Kryon_PDServer64BitSetup.exe.json` file.
2. Add the masking parameter:
   `"INSTALL_MASKING": "true"`
3. Make sure to save the changes
4. Run the installation as usual
5. After deploying and running Process Discovery, you can verify the Image-Masking installation.

Option 2: Post-Installation

1. **Installing the Required Components**:
   a. Open the file "`%ProgramData%\Kryon\installer-assets\config\prod\scripts\config.prod.properties.json`" to edit.
   b. Change the "`INSTALL_MASKING`" parameter to "`true`".
   c. Change the parameters `INSTALL_MONGODB` and `INSTALL_TESSERACT` to `True`.
   d. Save the changes.
   e. Open CMD as administrator and run:

```
C:\Kryon\installer-assets\config\prod\scripts
powershell .\configureAll.ps1 -h C:\Kryon -configDir "C:\Kryon\config" -n
prod -servicesDir "C:\Kryon\PDServer\MicroServices" -utilsDir
"C:\Kryon\PDServer\Support"
```

2. **Running Image-Masking through WatchDog**:
   a. Open the file "`%ProgramData%\Kryon\Services\Kryon.Server.ServicesWatchdog\appsettings.Production.json`" to edit.
   b. Search for the `Kryon.Service.Masking` section and change the `Enabled` parameter value to `True`.
   c. Save the changes.
   d. Restart the "Kryon Server - Process Discovery Service" windows service.
3. **Verifying Image-Masking Installation**:
   a. Open Seq and search for masking log messages http://localhost/seq/#/events?filter=masking



   b. Test the Masking API at Verify by calling http://localhost:50200/pd-masking/v1/isalive
4. **Enabling Image-Maskling**:
   a. Open the file `%ProgramData%\Kryon\config\prod\services\kryon-raw-fetcher-svc-default.json` to edit.
   b. Change the masking parameter to `true`
   c. Save the changes.
   d. Restart the "Kryon Server - Process Discovery Service" windows service.

Image Masking Scheduler

After enabling the Image Masking feature , you can set the **Image Masking Scheduler** to prepare the required images and mask them in advance, which is anytime before

you actually start reviewing the process results. This way the images are processed and masked in advance, without interfering with your work.

Scheduling Image Masking

By default, the image-masking scheduler is disabled. To schedule image-masking, all you need to do is edit some values in a dedicated JSON file and restart the PD service.

Enabling Image Masking:

1. Open the `C:\Kryon\config\prod\services\kryon-discovery-querifier-svc-default.json` file in an editor.
2. Locate the **ocrBakingScheduler** section and edit the values as needed:

```
"ocrBakingScheduler": {
  "enabled": false,
  "dayOfWeek": [0, 1, 2, 3, 4, 5, 6 ],
  "hour": 1,
  "minute": 0
}
```

Parameter	Value and description
enabled	`false` (default) : scheduler is disabled `true`: scheduler is enabled
dayOfWeek	set the days of the week the scheduler should run on (`0`-Sunday, `1`-Monday, etc.,)
Hour	What hour should the schedule run the image masking? `1` (default) Range: `1-24` (for example: 17 = 5 PM)
Minute	What minute in the already-set Hour should the scheduler run? Range: `1-60`

3. Save your settings
4. Restart the Process Discovery Service

Expected system behavior after setting the masking scheduler:

a. The discovery runs at the scheduled day and time
b. The results record is saved under recent discoveries (as any other discovery search)
c. The data collected as part of the scheduled discovery goes through a masking process

Accessing Logs

The following sections describe:

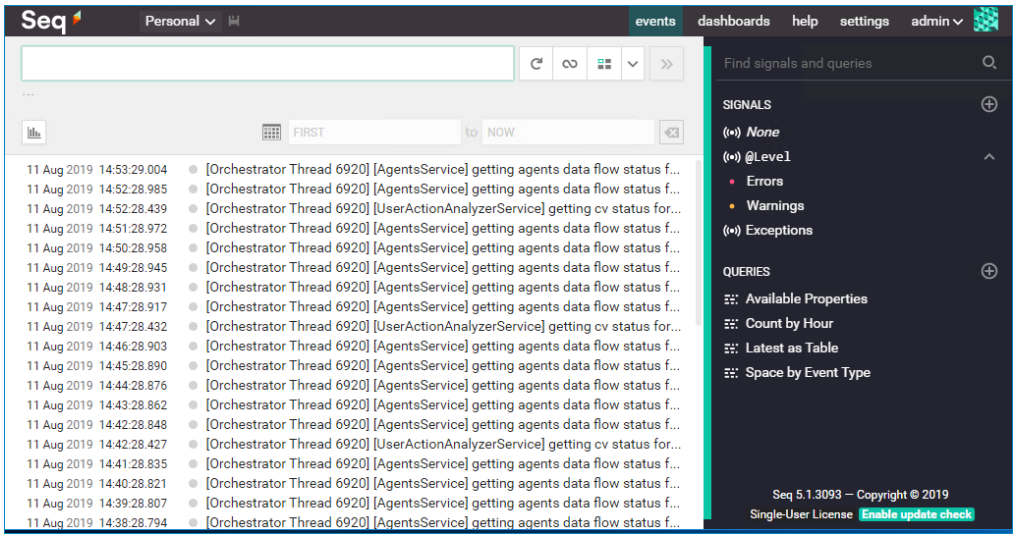
- Accessing Discovery Server Logs
- Accessing Discovery Robot Logs

Accessing Discovery Server Logs

As you work with Kryon Process Discovery, you will likely find it helpful to view the **Discovery Server** logs. To do so:

1. From a web browser with access to the **Discovery Server**, enter the following URL:
   `http://{Discovery Server IP address or FQDN (Fully Qualified Domain Name)}/seq`

   The Seq page opens, displaying the logs:



For more information about using Seq, see the Kryon Seq Tutorial at https://kryonsystems.force.com/KryoNet/s/article/Seq-Tutorial.

Accessing Discovery Robot Logs

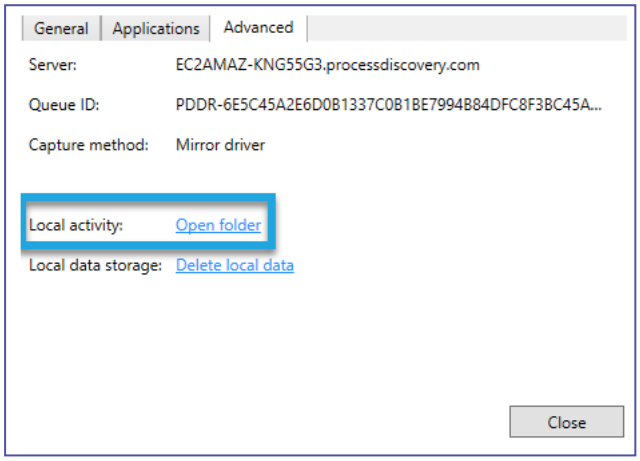
You can view **Discovery Robot** log files on the employee workstation, or you can download them directly to the **Discovery Server** without having to manage them from the client machines.

There are two log files:

- `pddr.log` contains debug and information level logs
- `pddr-Error.log` contains error logs

To view discovery robot log files on the employee workstation

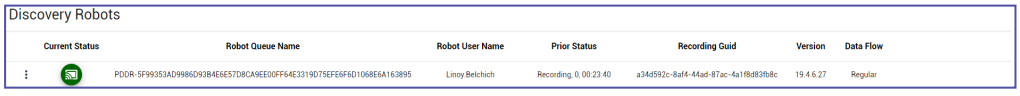
1. Right-click the robot's tray icon, and select **Kryon Process Discovery Options**.
2. From the Advanced Tab, click **Open Folder**



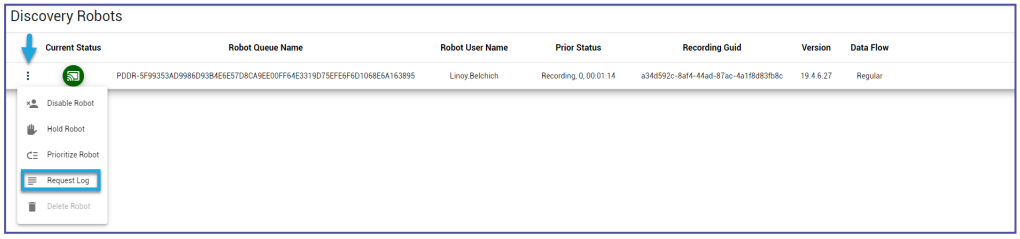
The Logs folder opens containing both the `pddr.log` and `pddr-Error.log` files

To download robot log files to the discovery server

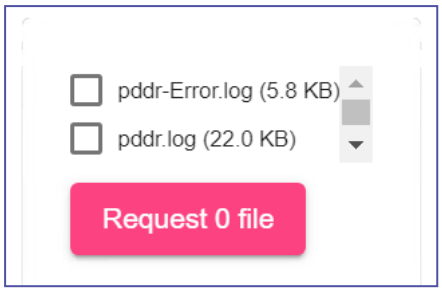
1. Access Kryon Process Discovery Admin
2. Click on **Robots**
3. The Discovery Robots screen opens:



4. Click the More (⋮) icon at the left of the robot's row and select **Request Log**



5. Select the file(s) you want to upload and click **Request [#] File(s)**



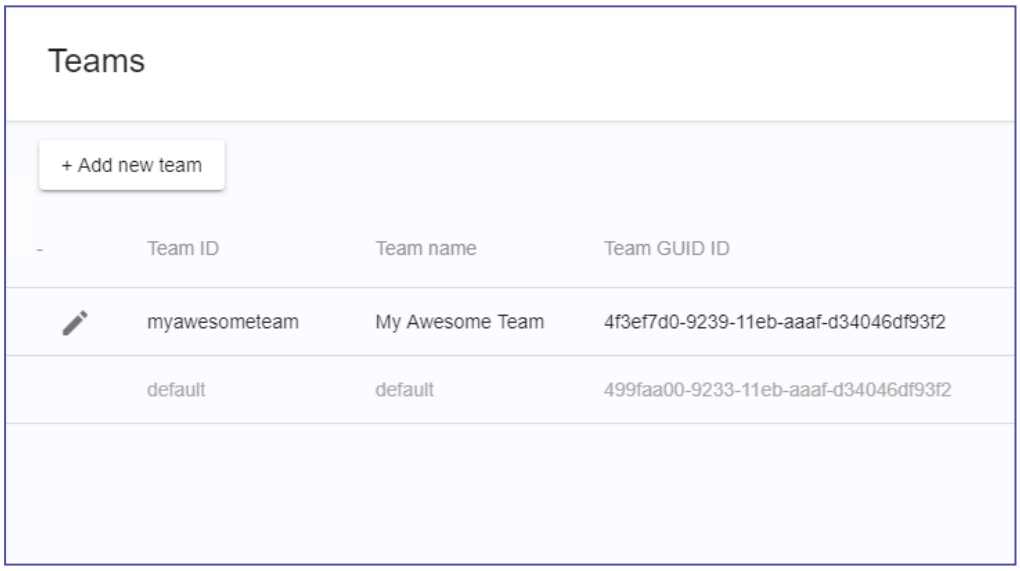
6. The logs are downloaded to the **Discovery Server**'s Download folder

Managing Recorded Sessions via Admin CLI

Getting the Team (Tenant) ID

Exporting/Importing recoded session is executed per Team ('tenant') only. To perform an export/import, you need to get the ID of the relevant Team first:

1. From the Process Discovery Console, go to **Settings**
2. Click **Teams**
3. Copy the **Team ID**



Exporting (uploading) recorded sessions data

1. Open CMD
2. Run and set the below:

```
cd C:\Kryon\PDServer\Support\kryon-admin-cli
set CONFIG_DIR=C:\Kryon\config
set NODE_ENV=prod
set KRYON_ENC_CFG=C:\Kryon\config\prod\general\kryon-decrypt.json
set NODE_PATH=C:\Kryon\PDServer\MicroServices\node_modules
```

3. Run 'node bin\cli.js' with the following parameters:

- For `--tenant`, update the relevant Team ID (tenant ID)
- For `--path`, update the uploaded raw data location in the machine

  EXAMPLE :

```
node bin\cli.js --tenant=714e6950-9518-11eb-85dc-ff8896fdb21c --
provider=rawData  --command=uploadRawDataToGW --
vaultAuthPath='C:/Kryon/PDServer/Support/pddr.keys' --path='C:/kryon/data'
```

NOTE

If you have a space in the folder-name from where you're copying the raw data, you might encounter an error in the command line.

- Optional parameters to modify and/or to add to the command:

  EXAMPLE: `--sendDelay=2000`

Parameter	Default value	Description	Alternative value
sendBatch	10	Batch actions size sent to RobotsGW	1-100
sendDelay	500	Delay in mill seconds for sending each batch	for actions more than 60k, use the value 2000
authProtocol	'http'	Authenticatio n protocol	'https'
authHost	file: `config/keycloak-token-client.json` { "keycloakHost" }	Aerobase Host	FQDN
authPort	file: config/keycloak-token-client.json { "keycloakPort" }	Aerobase Port	80

authRealm	file: config/keycloak-token-client.json { “keycloakRealmId“ }	Aerobase Realm	kryon
authClientId	pd-robot	Aerobase Client Id	pd-robot
authUser	pddr	Aerobase User	pddr
robotsGwQlPath	file: config/discovery.json { “robots-gw”: { “pathname” }}	robots Gateway Service GraphQL	/pddr/robotsgw/graphql
robotsGwHost	file: config/discovery.json { “robots-gw”: { “hostname” }}	robots Gateway Service Host	FQDN
robotsGwPort	file: config/discovery.json { “robots-gw”: { “port” }}	robots Gateway Service Port	80
robotsGwProtocol	http	Robots Gateway Service Protocol	https

Importing (downloading) recorded sessions data

1. Open CMD
2. Run and set the below:

```
cd C:\Kryon\PDServer\Support\kryon-admin-cli
set CONFIG_DIR=C:\Kryon\config
set NODE_ENV=prod
set KRYON_ENC_CFG=C:\Kryon\config\prod\general\kryon-decrypt.json
set NODE_PATH=C:\Kryon\PDServer\MicroServices\node_modules
```

3. Run '`node bin\sli.js`' with the following parameters:
   - For `--tenant`, update the relevant Team ID (tenant ID)
   - For `--path`, update the location to which you want to save the downloaded data

NOTE

If you have a space in the folder-name from where you're copying the raw data, you might encounter an error in the command line.

EXAMPLE:

```
node bin\cli.js  --tenant=86b6a8e0-67a3-11eb-a999-23eb75c380fb  --
provider=rawData  --command=downloadRawData   --dbProviderName=mongodb  --
path='C:/Temp'
```