KRYON

BE YOUR FUTURE

Installation & Administration Guide

Kryon Process Discovery v19.1

This document contains Kryon Systems proprietary information. The information contained herein is confidential and cannot be distributed without the prior written approval of Kryon Systems Ltd.

© 2008-2019 Kryon Systems Ltd.
All rights reserved.

Document revision: 12-Feb-2019

Contents

APPENDIX C: Enhanced Security Options for Discovery Console

CHAPTER 1: Introduction

Welcome to Kryon Process Discovery – a powerful, proprietary, AI-based platform designed to identify your organization's business processes, correlate variants, and make recommendations for enhanced efficiency via automation.

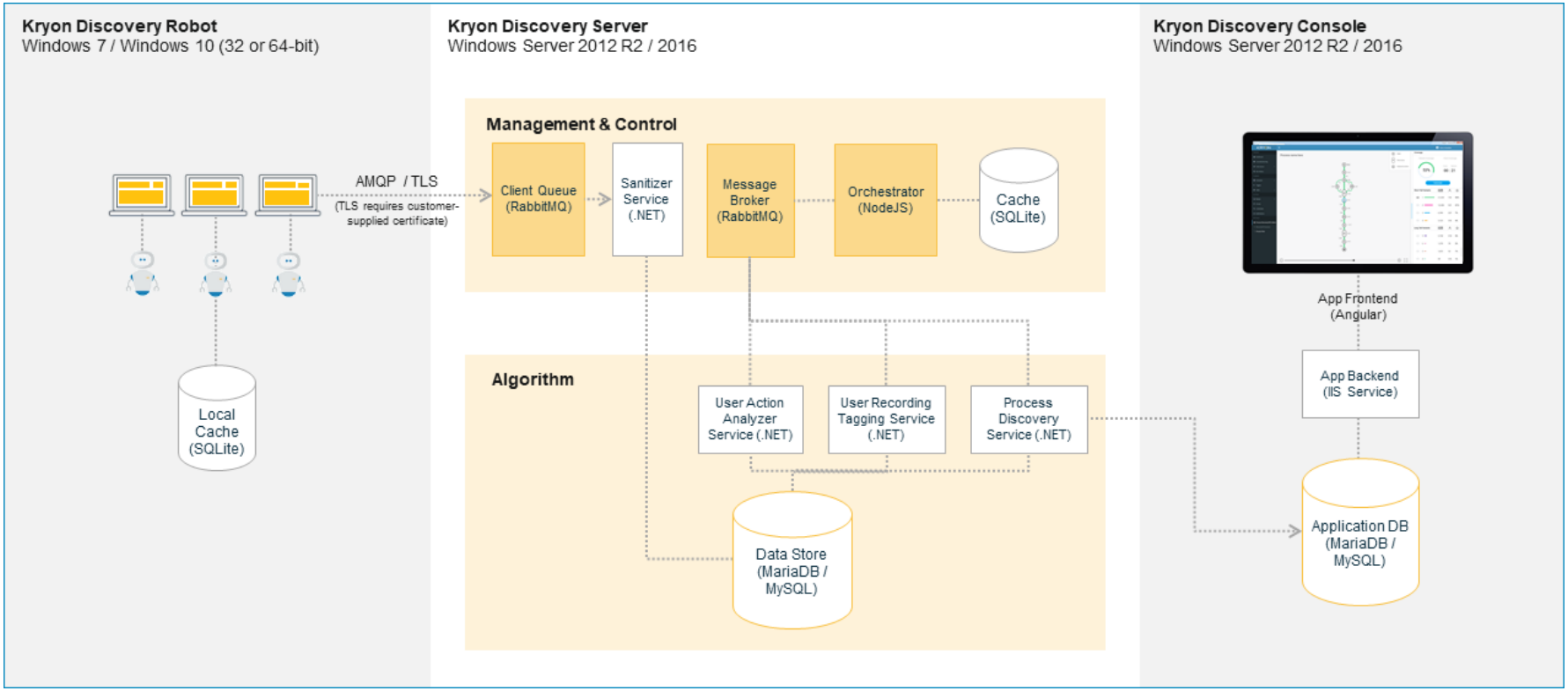
This guide explains the steps required to install the Discovery Server and Discovery Robots. It also includes information about administering the Process Discovery platform (including managing Process Discovery users and credentials, and blacklisting/whitelisting applications and websites to be monitored by Discovery Robots).

Intended audience

The Kryon Process Discovery platform is an enterprise system involving multiple components and numerous networking/security considerations. This guide is intended for IT professionals with a general knowledge of Windows (server and desktop editions), database management, and networking and security protocols.

In this chapter:

System Architecture



Process Discovery 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** is sent to the **Discovery Server** for analysis.

Discovery Database

The **Discovery Database** is the database (either MariaDB or MySQL) in which all the data collected by the **Discovery Robots** is stored. The data collected by the **Discovery Robots** is transferred almost immediately to the **Discovery Database** and remains on the client machine for only very short time.

Discovery Server

The data stored in the **Discovery Database** is utilized by the **Discovery Server**, where it undergoes a complex algorithmic process, including:

- Computer vision algorithm – extraction of relevant information from every image
- Tagging algorithm – identification of each individual action on each screen and assignment of a unique tag to each, facilitating recognition and matching of repeated actions
- Process discovery (machine learning) algorithm – comparison and compilation of extracted and tagged information; mapping of processes and variants
- Automation recommendation engine – calculation of automation recommendations
- Output of process and variant data to the **Discovery Console**

Process Discovery's AI mechanism, as executed by the **Discovery Server**, gets smarter and more effective as more and more data is gathered by the **Discovery Robots.**

Discovery Console

A browser-based application providing management an overview of discovered processes, with the ability to drill down into all the underlying details. The **Discovery Console** presents real-time, visual maps of each process and all its different variations, allowing managers to visualize how each activity, application, and human interaction relates to process efficiency.

The **Discovery Console** provides a quick and convenient interface for sending processes directly to **Kryon Studio** as pre-developed automation workflows.

Studio Integration

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

The integration between the **Discovery Console** and **Studio** allows managers to send processes directly to automation as pre-developed workflows, including workflow 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.

System Requirements

	Discovery Server	Discovery Robot	
		minimum	recommended
Machine role(s)	• Machine learning • Data processing • Data storage • Console	Monitored desktop	
OS	Windows Server 2012 R2 or higher	• Windows 10 (64- or 32-bit); ***or*** • Windows 7 (64- or 32-bit)	
Database	• MariaDB 10.3.7 or higher; ***or*** • MySQL 8.0.11 or higher	N/A	
Software Prerequisites	.NET Framework 4.7.1 (should be installed prior to Discovery Server installation)	.NET Framework 4.7.1 (will be installed by Discovery Robot installation package if not previously installed)	
Processor (minimum) *(See How many cores? for details about how to verify the number of processor cores)*	Intel i7 or Xeon 16 cores **NOTE:** 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 configuration details if this is option is required.*	i3 2 cores	i5 4 cores
RAM (minimum)	32GB	4GB	
Free disk space	500GB SSD	50GB HDD	
Firewall	Open firewall ports: • 5672 (or 5671 when using TLS) – customizable • 8080 (or 443 when using TLS) – configurable through IIS	N/A	

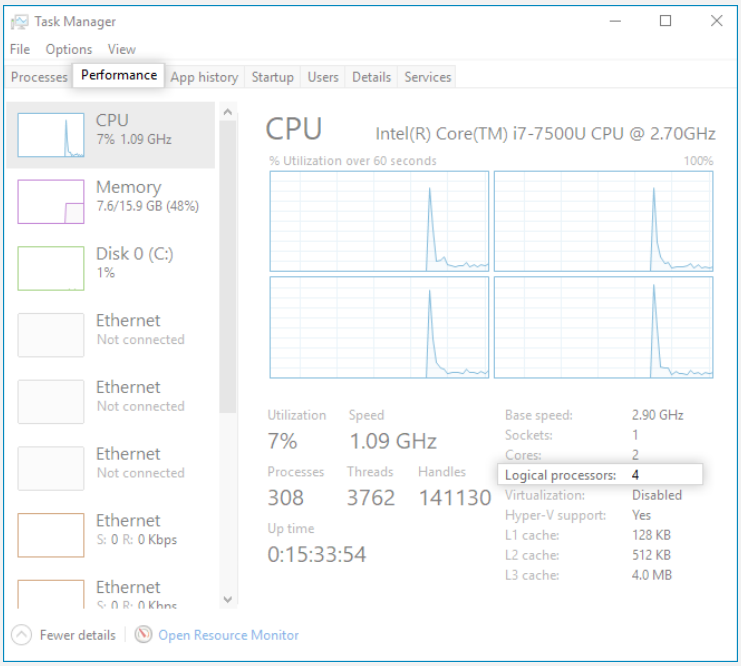
	Discovery Server	Discovery Robot	
		minimum	recommended
Network bandwidth/day	500 MB/day per active Discovery Robot client	N/A	
Network bandwidth/second	~15 KB/s per active Discovery Robot client	N/A	
Additional	Supported browser for accessing Discovery Console: • Chrome v69 or higher Installation on a dedicated physical server is highly recommended. However, use of a VM server is possible when this is not an option. The VM should: (1) meet the specifications listed above; and (2) must have a dedicated (i.e., non-shared) CPU.	Supported browsers for Discovery Robot recording: • Chrome v69 or higher • Edge v17 or higher • Internet Explorer v11 or higher	

NOTE

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 counterintuitive, but for purposes of Process Discovery, it's the **Logical processors** field you're interested in – not the **Cores** field!


Task Manager
File Options View
Processes Performance App history Startup Users Details Services
CPU
7% 1.09 GHz
Memory
7.6/15.9 GB (48%)
Disk 0 (C:)
1%
Ethernet
Not connected
Ethernet
Not connected
Ethernet
Not connected
Ethernet
S: 0 R: 0 Kbps
Ethernet
CPU
Intel(R) Core(TM) i7-7500U CPU @ 2.70GHz
% Utilization over 60 seconds
100%
Utilization
7%
Speed
1.09 GHz
Processes
308
Threads
3762
Handles
141130
Up time
0:15:33:54
Base speed: 2.90 GHz
Sockets: 1
Cores: 2
Logical processors: 4
Virtualization: Disabled
Hyper-V support: Yes
L1 cache: 128 KB
L2 cache: 512 KB
L3 cache: 4.0 MB
Fewer details
Open Resource Monitor

CHAPTER 2: Installing the Kryon Process Discovery Server

In this chapter:

Discovery Server Installation Steps

NOTE

Get connected!

During installation of the **Discovery Server**, the machine on which you are installing must be connected to the Internet. (Certain files are automatically downloaded from Microsoft in order to install the IIS Server component.) Once installation is complete, the server can be disconnected from the Internet.

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

1. Open firewall ports
2. Ensure that the server machine meets the required hardware specifications
3. Install .NET Framework 4.7.1
   - Reboot the machine if prompted to do so
4. Install MariaDB 10.3.7 or higher *(recommended)* ***or*** MySQL 8.0.11 or higher (either on the **Discovery Server** machine or on a remote machine)
   - When installing the database, be sure to tick the checkboxes for: (1) enabling access from remote machines for the root user and; (2) using UFT8 as the default server's character set
   - If you are installing the database on a remote machine, be sure to:
     - change the default value of the DB_SERVER parameter when running the installation package (step 5); and
     - make the required configuration changes referenced in step 8a
5. Run the Process Discovery Server installation package
6. If you are using MySQL, follow additional database configuration steps
7. From the **Windows Services** app or the **Windows Task Manager > Services** tab, restart the **MySQL** service
   - **NOTE:** Yes, even if you are installing MariaDB, the name of the service that needs to be restarted is MySQL
8. Set up the Kryon Process Discovery license
9. Apply additional configuration options (as required in the following scenarios):
   a. Database & Discovery Server on different machines
   b. Non-default port for communication between Discovery Robots & Discovery Server

10. Configure TLS on the server side if you will be using it to secure communication between the **Discovery Robots** and the **Discovery Server**

11. (Recommended) As you work with Process Discovery, you may find it helpful to view the server logs. To enable this capability:

    - Download **Seq** from this location: https://getseq.net/Download
    - Install and configure **Seq** using the default options provided in the installation package

NOTE

Accessing the server logs

After installing **Seq**, you can access the server logs as follows:

- From a web browser with access to the **Discovery Server**, enter the following URL:

  `http://{serverIP}:5341`

  - In the above URL, replace `{serverIP}` with the actual IP address of the **Discovery Server**
  - If you are accessing the logs from the **Discovery Server** machine itself, replace `{serverIP}` with `localhost`

12. (Optional) If required by your organization's security policies, implement enhanced security options for **Discovery Console** by adding custom HTTP response headers

Open Firewall Ports

Follow these steps to prepare your network for Kryon Process Discovery installation:

1. Open a port for communication between the **Discovery Robots** and the **Discovery Server**. Default configuration is as follows:
   - **5672** when not using TLS; ***or***
   - **5671** when using TLS

NOTE

Customizing the default port settings

To customize these default port settings, open the desired port in your network firewall and the Windows firewall; then follow these steps:

- For non-TLS installations
- For TLS installations

2. Open a port for communication between the **Discovery Console** and web browser. Default configuration is as follows:
   - **8080** when not using TLS; ***or***
   - **443** when using TLS

NOTES

Customizing the default port settings

To customize these default port settings, open the desired port in your network firewall and the Windows firewall; then configure the port settings in IIS.

If removing the default website from IIS

By default, **Discovery Server** installation does not remove the entry for `default website` from IIS. If you wish, you can do so by adding a parameter when running the Process Discovery Server installation package. Note that if do choose to do this, the default port for communication between the **Discovery Console** and web browser will be set to **80** (as opposed to **8080**).

3. (Optional) If you want the ability to access Orchestrator and/or the server logs remotely (i.e., from a machine other than the **Discovery Server** itself), open the following ports:
   - **8788** for Orchestrator
   - **5341** for server logs

CAUTION

So long as you will be accessing Orchestrator and the server logs directly from the **Discovery Server**, opening ports 8788 and 5341 is not required in order for Process Discovery to work properly.

The applications behind these ports (8788 and 5341) contain sensitive data, so do consider the risks before opening them.

Run the Process Discovery Server Installation Package

The **Discovery Server** installation is a silent installation performed from an elevated command prompt (i.e., the Windows command prompt **run as administrator**). In addition to installing the core **Discovery Server** files, the installation package also automatically installs the following third-party components:

- RabbitMQ Server (the queue manager for communications between the **Discovery Robots** and the **Discovery Database**)
- Erlang OTP (the programming language on which RabbitMQ is built)
- NodeJS (JavaScript runtime used by Orchestrator, Process Discovery's administration tool)

CAUTION

Silence is golden...

Installation will fail if not run in silent mode, so pay special attention to the following:

- **DO** include the `-silent` switch in the command line
- **DO NOT** attempt to install by double-clicking the EXE file

Exit code

To view an exit code following installation:

1. Install using the **START** command with the **/WAIT** option in the command line, as follows:

   ```
   START /WAIT {InstallerLocation}\PDServer64BitSetup.exe -silent
   ```

2. Immediately after installation completes, run the **ECHO** command with the **%errorlevel%** parameter, as follows:

   ```
   ECHO %errorlevel%
   ```

Possible exit codes

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

Installation parameters

When installing the **Discovery Server**, installation parameters can be specified, either:

- in the command line; ***or***
- in the JSON file included in the installation package

Parameters must be specified using exact parameter names (case-sensitive).

NOTE

JSON overrides the command line

In case of inconsistencies between the parameters specified in the command line and the JSON file, the values specified in the JSON file will override those specified in the command line.

The following installation parameters are supported:

Parameter Name	Default Value (if unspecified/left blank)	Description
DB_ROOT_PASSWORD	**NO default value** ** mandatory – installation will fail if not specified	Password used by the **Discovery Server** to access the database (as a superuser)
DB_SERVER	`localhost`	IP address/machine name/FQDN (Fully Qualified Domain Name, i.e., DNS 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 ◦ For additional configuration changes required post-installation in this scenario, see Database & Discovery Server on Different Machines
DB_USER_NAME	hard coded; should not be modified	Username used by the **Discovery Server** to access the database

Parameter Name	Default Value (if unspecified/left blank)	Description
DB_USER_ PASSWORD	hard coded; should not be modified	Username used by the **Discovery Server** to access the database
InstallFolder	`{Local drive with the most free space}:\Program Files\Kryon Process Discovery Server`	Folder in which the core **Discovery Server** files will be installed **SYNTAX:** The syntax for the InstallFolder value varies by whether you are specifying it using the command line or the JSON file: • In the command line – `X:\InstallFolder` (as it appears in the Windows Explorer navigation bar) • In the JSON file – `X:\\InstallFolder` (double backslash in each location in which Windows syntax would use a single backslash) **NOTES:** • The InstallFolder must be on a local drive • If the drive on which the InstallFolder is located does not contain enough free space for installation, the installation will fail • The InstallFolder cannot be the root folder of a drive (e.g., C:\) • All components will be installed parallel to the specified InstallFolder ◦ For example, if `InstallFolder="X:\Products Folder\PD"` then the RabbitMQ component will be installed to `"X:\Products Folder\RabbitMQ"`, etc.

Parameter Name	Default Value (if unspecified/left blank)	Description
REMOVE_DEFAULT_WEBSITE	`false`	By default, **Discovery Server** installation does not remove the entry for `default website` from IIS. If you wish to do so, add this parameter either to the command line or to the JSON file and set its value to `true`. Note that this will also change the default port for communication between the **Discovery Console** and web browser to **80** (as opposed to **8080**).

CAUTION

Database root password is required

- **DB_ROOT_PASSWORD** is a required parameter and must be specified either in the command line or in the JSON file.
- All other supported parameters are optional. If left unspecified, their default values will be used.

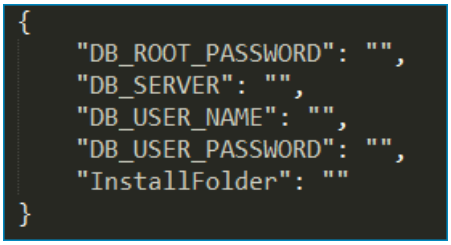
Specifying parameters in the command line

The command line would look like this when parameters are specified:

- When using START /WAIT:
```
START /WAIT {InstallerLocation}\PDServer64BitSetup.exe [Parameter1=Value1 Parameter2=Value2 Parameter3=Value3] -silent
```
- When not using START /WAIT:
```
{InstallerLocation}\PDServer64BitSetup.exe [Parameter1=Value1 Parameter2=Value2 Parameter3=Value3] -silent
```

Specifying parameters in the JSON file

The **Discovery Server** installation package includes a JSON file that you can use to specify installation parameters if you prefer (instead of in the command line):



If you if you elect to use the JSON file, follow these steps prior to running the installation package:

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

   `"DB_ROOT_PASSWORD": "pdpassword1234!",`

   - Note that the `REMOVE_DEFAULT_WEBSITE` parameter does not appear by default in the JSON file. If you will be using this parameter, add it manually on a separate line anywhere between the opening and closing `{ }`
3. Save the JSON file
   - Ensure that it is named: **PDServer64BitSetup.exe.json**
   - Ensure that it is located in the same folder as the **PDServer64BitSetup.exe** file

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 will open to the logged-in user's `Temp` folder
   - If you are logged into the machine as Administrator, the logs will be located in this folder
   - If you logged into the machine as a user other than Administrator, the logs will be located one directory level up

NOTE

Time is of the essence...

The filename of each installation log includes a timestamp. Note that this timestamp will be the same for each log created during a single run of the **Discovery Server** installation package.

Additional Steps for Configuring MySQL

Follow these steps only if you are using MySQL as your database:

1. From the **Windows Services** app or the **Windows Task Manager > Services** tab, restart the **MySQL** service
2. Run the CMD prompt as an administrator
3. Change the directory to `{MainPDFolder}\Server\Pipe\resources\dbscripts\mysql`
4. Run the following commands from the command prompt:
   - `type drop.sql | mysql -h localhost -u root --password=DB_ROOT_PASSWORD --default-character-set=utf8mb4`
   - `type create.schema.users.mysql.sql | mysql -h localhost -u root --password=DB_ROOT_PASSWORD --default-character-set=utf8mb4`
     - In the above commands, replace **DB_ROOT_PASSWORD** with the value you specified when running the Discovery Server installation package – either as a command-line parameter or in the JSON file
       - Recall that **DB_ROOT_PASSWORD** was a required parameter in order for installation to succeed

Set Up the Kryon Process Discovery License

Follow these steps to set up your Process Discovery license:

1. Rename the license file you received with the name `license.llk`
2. Copy the file to the folder in which the core **Discovery Server** files were installed
   - The default folder location is `C:\Program Files\Kryon Process Discovery Server\` (assuming `C` was the local drive with the most free space at the time of installation)
   - If you specified a different InstallFolder location during installation, this the location to which you should copy the license file now
3. From the **Windows Services** app or the **Windows Task Manager > Services** tab:
   - Restart the **PD_Orchestrator** service
   - Restart the **PD_Pipe** service
   - Check the **Status** column of the **PD_Orchestrator** and **PD_Pipe** services to ensure that both are `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.

CHAPTER 3: Installing Discovery Robots

In this chapter:

Discovery Robot Installation Steps

NOTES

Two options for running the Discovery Robot installation package

The Discovery Robot installation package supports two options for installation:

1. UI mode (invoked by double-clicking on the package's **EXE** file and following the prompts; ***or***
2. Silent mode (which allows you to specify certain installation options through parameters in the command line)

Select the option most appropriate for your situation and follow the instructions in the relevant topic.

.NET Framework 4.7.1

The Discovery Robot installation package will check for the presence of .NET Framework 4.7.1. If it is not installed (or if a lower version of .NET is installed), the installer will install it automatically.

RECOMMENDED

Install the Process Discovery Server first

Follow the steps for installing the Process Discovery Server prior to installing **Discovery Robots**.

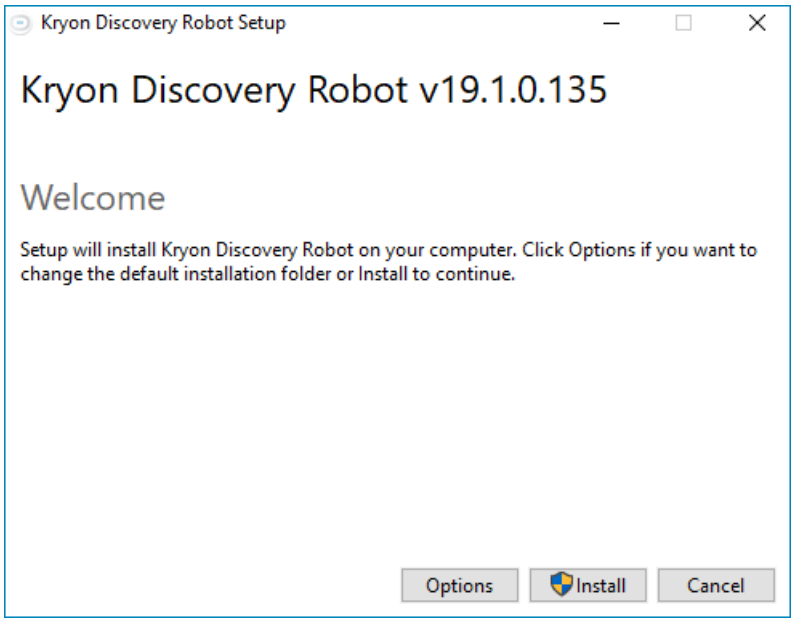
Follow these steps to install a **Discovery Robot** on each client machine:

1. Ensure that the client machine meets the required hardware specifications
2. Run the **Discovery Robot** installation package either in UI mode or silent mode
3. Configure the **Discovery Robot** with the IP address of the **Discovery Server**
   - **NOTE:** This step is not necessary if you installed the **Discovery Robot** in silent mode and specified the **Discovery Server's** IP address in the `messagesBrokerHost` command line parameter
4. Follow these instructions if you are using a non-default port for communication between the **Discovery Robots** and the **Discovery Server**
5. Configure TLS on the robot if you will be using it to secure communication between the **Discovery Robots** and the **Discovery Server**
6. Restart the robot
   - Right-click the robot's tray icon, and select **Exit**
   - Click the **Windows Start** button , type `pddr.exe` and select the file to run it

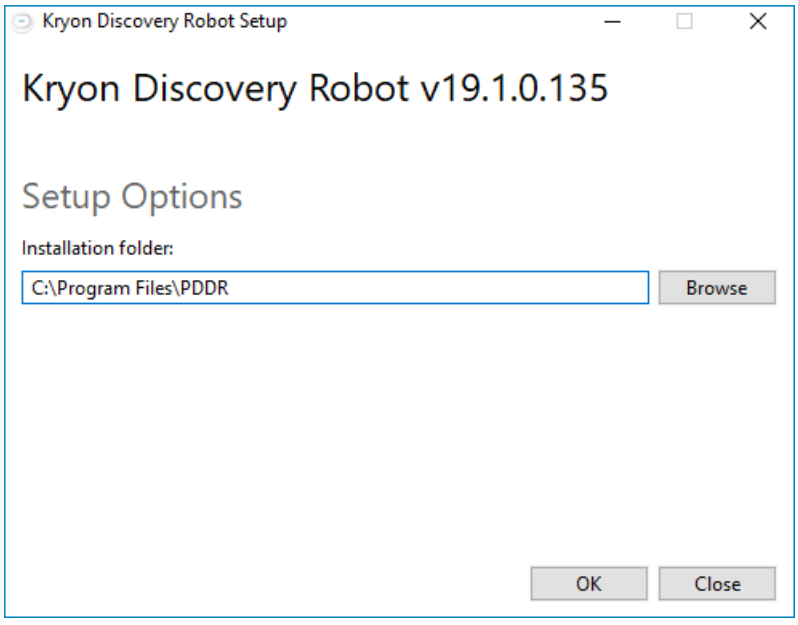
UI Mode Installation

Follow these steps to install a **Discovery Robot** in UI mode:

1. Right-click the file `DiscoveryRobot64BitSetup.exe`, and select **Run as administrator**
2. The Kryon Discovery Robot Setup **Welcome** screen will open:

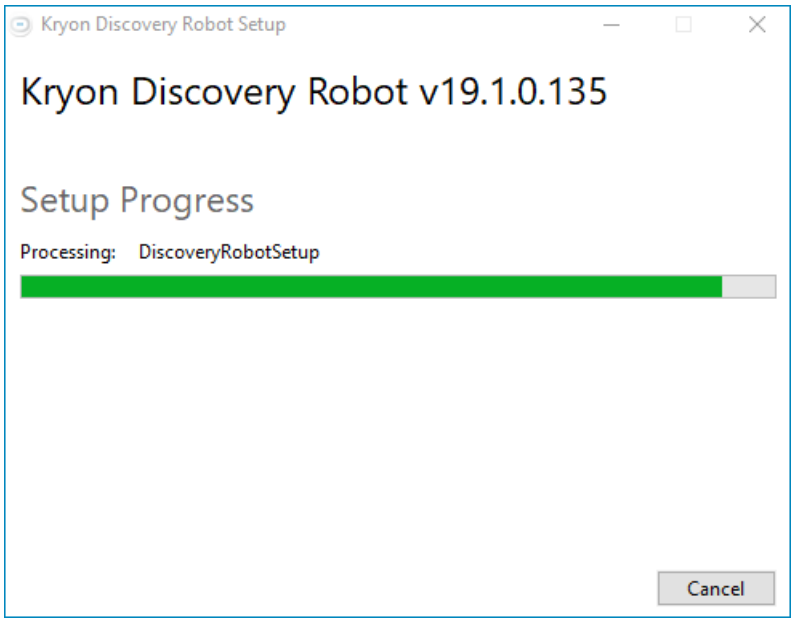


3. If you wish to change the default installation directory, click **Options**, then set the desired directory in the **Setup Options** screen:

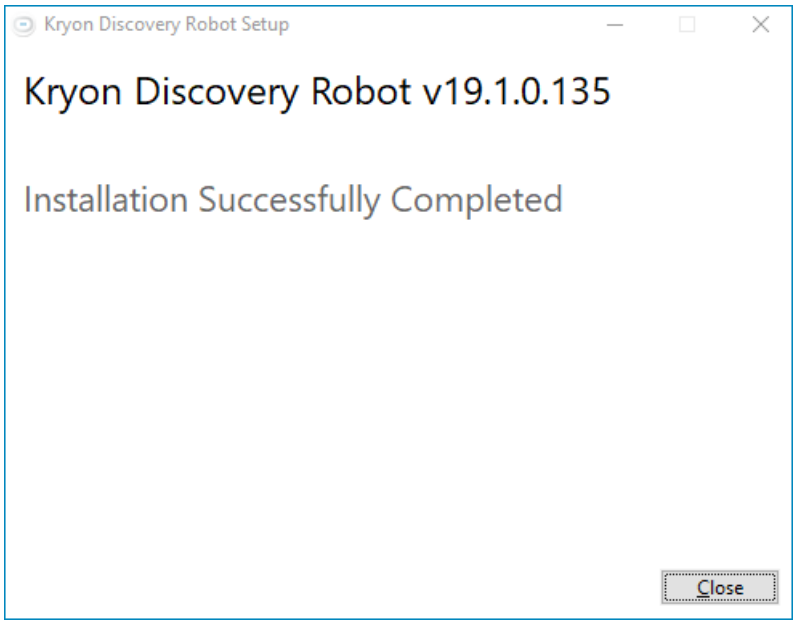


4. Click the **Install** button to install the **Discovery Robot**

5. The progress bar will indicate that installation is proceeding:



6. You will see the following screen when installation is complete:



7. Continue by configuring the **Discovery Server** address and additional **Discovery Robot** options as required

Silent Mode Installation

Run the **Discovery Robot** installation in silent mode from an elevated command prompt (i.e., the Windows command prompt **run as administrator**).

CAUTION

Silence is golden...

- Be sure to include the `-silent` switch in the command line or the installer will open in UI mode.

Exit code

To view an exit code following installation:

1. Install using the **START** command with the **/WAIT** option in the command line, as follows:

   ```
   START /WAIT
   {InstallerLocation}\DiscoveryRobot64BitSetup.exe -silent
   ```

2. Immediately after installation completes, run the **ECHO** command with the **%errorlevel%** parameter, as follows:

   ```
   ECHO %errorlevel%
   ```

Possible exit codes

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

Installation parameters

Silent mode installation supports the inclusion of the following installation parameters in the command line:

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`	IP address/machine name/FQDN (Fully Qualified Domain Name, i.e., DNS 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 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.

Specifying parameters in the command line

The command line would look like this when parameters are specified:

- When using START /WAIT:
```
START /WAIT
{InstallerLocation}\DiscoveryRobot64BitSetup.exe
[Parameter1=Value1 Parameter2=Value2 Parameter3=Value3] -
silent
```
- When not using START /WAIT:
```
{InstallerLocation}\DiscoveryRobot64BitSetup.exe
[Parameter1=Value1 Parameter2=Value2 Parameter3=Value3] -
silent
```

Parameters must be specified using exact parameter names (case-sensitive)

Configuring Discovery Robot Options

To change **Discovery Robot** options following installation, open the file `{MainRobotFolder}\pddr.exe.config` with a text editor and edit the following keys:

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

Original parameter	Description/change
`<add key="messagesBrokerHost" value="localhost"/>`	**NOTE:** This parameter sets the address of the **Discovery Server** `<add key="messagesBrokerHost" value="{Discovery Server IP address/machine name/FQDN (Fully Qualified Domain Name, i.e., DNS name)}"/>` • **MUST** be changed from the default value of `localhost` ◦ The only case in which this key does not need to be changed is if the actual IP address of the **Discovery Server** was specified in the `messagesBrokerHost` command line parameter during a silent installation
`<add key="stealthMode" value="false"/>`	• If you want the Discovery Robot's tray icon to be visible to the machine's end user(s) → **no change** • If you want the Discovery Robot's tray icon to be hidden → change to `<add key="stealthMode" value="true"/>`
`<add key="persistRecords" value="false"/>`	• If you want to maintain logs of Discovery Robot activity for debugging purposes → change to `<add key="persistRecords" value="true"/>`

Installation Log

The **Discovery Robot** installer (both UI mode and silent mode) records a detailed logs of the entire installation process. This log can be a useful resource for troubleshooting.

To locate the log:

1. Right-click the **Windows Start** button
2. Select **Run**
3. Type `%temp%`, then hit <ENTER>
4. A Windows Explorer window will open to the logged-in user's `Temp` folder, in which you will find the log (look for a file with the name `PDDRClient64BitSetup` – along with a version number and a timestamp)

CHAPTER 4: Process Discovery Administration

In this chapter:

Accessing Orchestrator

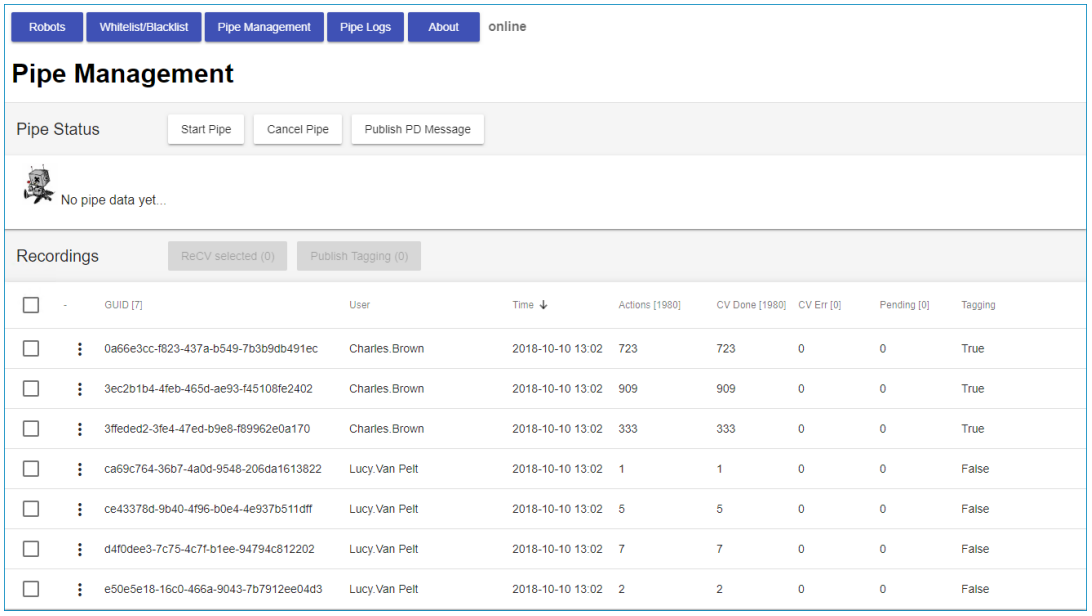
The Process Discovery administration tool is known as **Orchestrator**. Access it as follows:

1. From a web browser with access to the **Discovery Server**, enter the following URL:

   `http://{serverIP}:8788/app/`

   - In the above URL, replace `{serverIP}` with the actual IP address of the **Discovery Server**
   - If you are accessing **Orchestrator** from the **Discovery Server** machine itself, replace `{serverIP}` with `localhost`

2. The following screen will open:



NOTE

No need to worry about every screen and button!

Much of the **Orchestrator** app is intended primarily for support use. Therefore, only certain screens and procedures are documented in this guide. Any features not documented should be used only as instructed by the Kryon Support Team.

Configuring a Whitelist and/or a Blacklist

The whitelist and/or blacklist determines exactly which applications will be monitored by the **Discovery Robots**.

- The whitelist is a list of applications/websites that will always be monitored
- The blacklist is a list of applications/websites that will **NOT** be monitored)

NOTES

Black trumps white

Recommended practice is to configure either a whitelist or a blacklist (but not both). However, in cases where both exist, if there are conflicts, the

blacklist overrides the whitelist.

- **Example:** If LinkedIn appears on both the whitelist and the blacklist, LinkedIn will **NOT** be monitored since the blacklist takes precedence over the whitelist.

Nothing means everything

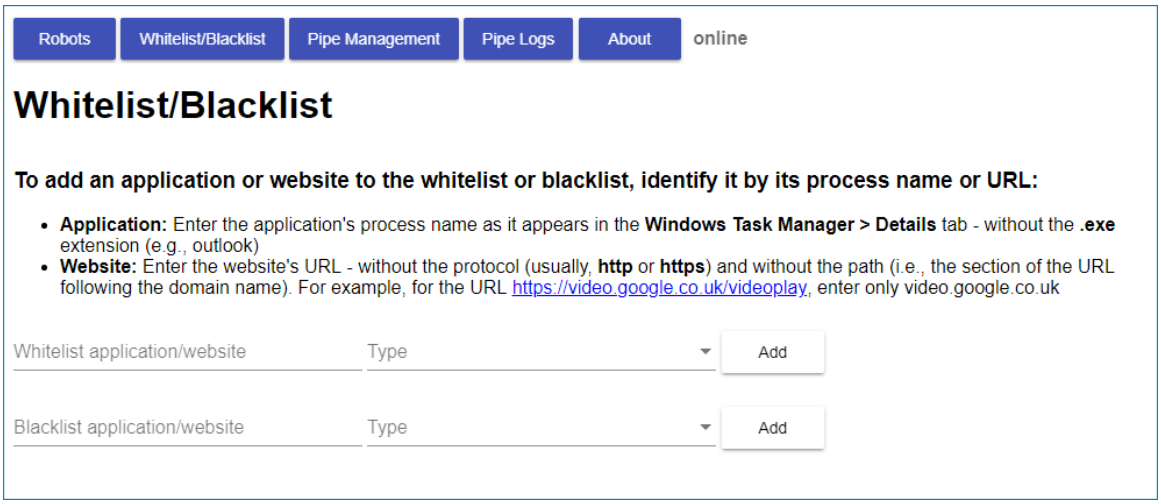
- If there is no whitelist and no blacklist, the **Discovery Robots** will monitor and record **ALL** applications and websites

Accessing the whitelist/blacklist in Orchestrator

To access the whitelist/blacklist in **Orchestrator**:

1. Access Orchestrator
2. Click the **Whitelist/Blacklist** tab

The following screen will open:



Adding an application to the whitelist/blacklist

To add an application or website to the whitelist/blacklist:

1. Click the field for the list to which you want to add the application/website
2. For an application:
   a. Enter the application's process name as it appears in the **Windows Task Manager > Details** tab – without the `.exe` extension (e.g., `outlook`)
   b. From the **Type** dropdown list, select **Process name**
3. For a website:
   a. Enter the website's URL – without the protocol (usually, `http` or `https`) and without the path (i.e., the section of the URL following the domain name)
      - For example, for the URL `https://video.google.co.uk/videoplay`, enter only `video.google.co.uk`
   b. From the **Type** dropdown list, select **url**
4. Click the Add button

The application/website will now appear in the relevant list:

Deleting an application from the whitelist/blacklist

To delete an application from the whitelist/blacklist:

1. Click the **X** next to the name of the application/website in the list

Managing User Credentials

To manage user credentials for accessing the **Discovery Console**:

1. Open the following file with a text editor: `C:\Program Files\Kryon Process Discovery Server\Console\Kryon.ABPD.API\Web.config`
2. Edit the value in the key that reads: `<add key="LoginAuthorizedUsers" value="" />` to add/delete user credentials, using this format:

   `username1@password1;username2@password2;username3@password3`

EXAMPLE

Process Discovery user credentials

Assume you want to authorize 2 users for **Discovery Console**:

- The first user's username is `ALincoln`, and his password is `4score7years`
- The second user's username is `HTruman`, and his password is `BuckStopsHere`

The relevant key should read:

```
<add key="LoginAuthorizedUsers"
value="ALincoln@4score7years;HTruman@BuckStopsHere" />
```

Starting the Pipe

When you look at Kryon Process Discovery from the highest level, there are 2 main phases of data collection and processing:

- **Recording –** the **Discovery Robots** monitor the usage of business applications on client machines, and the data is transferred to the **Discovery Server** and stored in the database
- **Pipe –** the **Discovery Server** works its algorithmic magic: extracting, analyzing, identifying, tagging, comparing, compiling, mapping, recommending... then it outputs the formatted data to the **Discovery Console**

Once you install the **Discovery Robots** and start them running, the recording phase begins.

But when and how often to run the pipe phase depends very much on your organization and the volume of data collected by the **Discovery Robots**. So, there are 2 different options for starting the pipe:

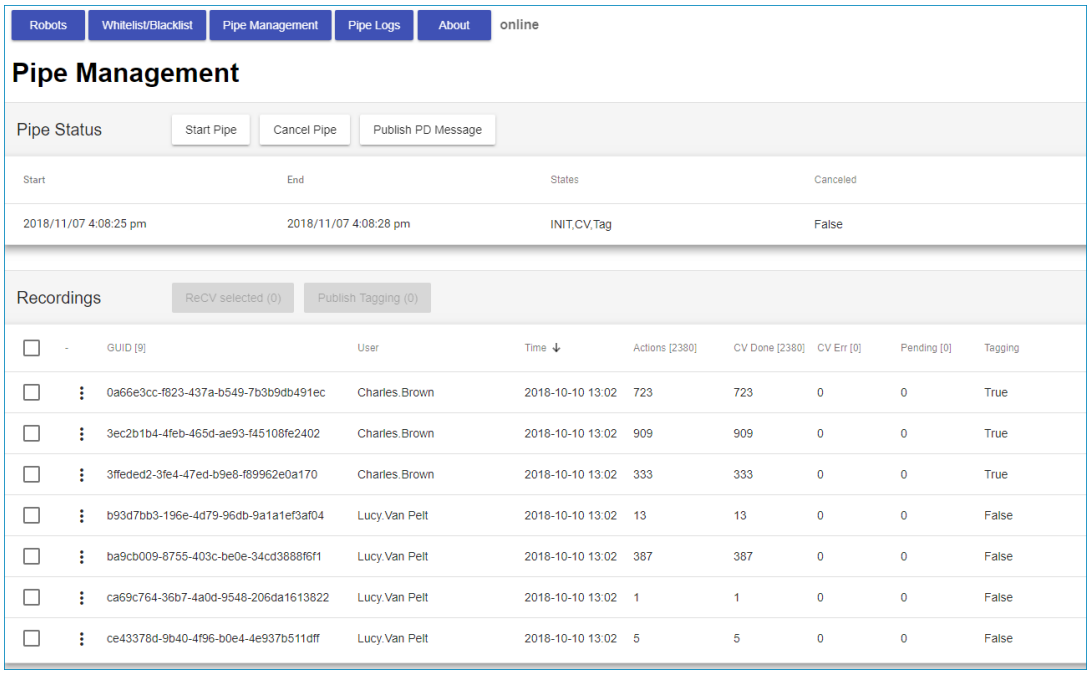
- Manually running the pipe
- Scheduling the pipe

Manually running the pipe

To manually run the pipe:

1. Access Orchestrator
2. Click the 

tab
3. The following screen will open:



4. Click the Start Pipe button to start the pipe

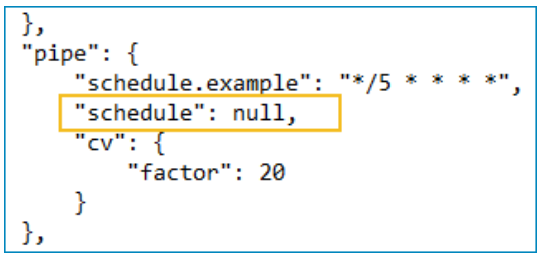
Scheduling the pipe

Depending on the length of the Process Discovery cycle your organization wants to adopt, you can schedule the pipe to run automatically at intervals you specify.

To schedule the pipe:

1. Open the following file with a text editor:
   `{MainPDFolder}\Server\Orchestrator\config\production.json`
   - **NOTE:** In path above, `{MainPDFolder}` = the folder in which the core **Discovery Server** files were installed
     - By default, this folder is `C:\Program Files\Kryon Process Discovery Server\` (assuming `C` was the local drive with the most free space at the time of installation)
     - If you specified a different InstallFolder location during **Discovery Server** installation, `{MainPDFolder}` is the InstallFolder you specified

2. Edit the line under `"pipe"` that reads `"schedule": null` by replacing `null` value with a **cron** expression (inside double quotes) representing the schedule you want to adopt. Learn more about how to create **cron** expressions here.

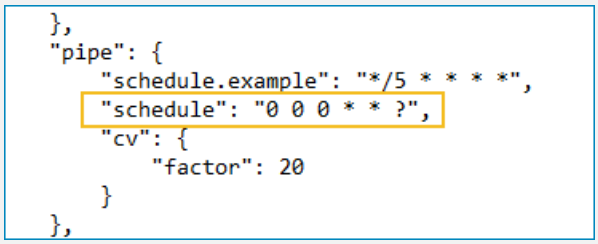


EXAMPLE

Scheduling the pipe using a cron expresion

Assume you want to schedule the Process Discovery pipe to run every night at midnight.

- The cron expression representing `every night at midnight` is `0 0 0 * * ?`
- The `pipe` section of `{MainPDFolder}\Server\Orchestrator\config\production.json` would look like this:



Deleting Data

After you have run Process Discovery for a time and your Business Process Analyst has already analyzed the data and exported relevant processes as automations, best practice would suggest deleting existing data from the database and starting again from the beginning – perhaps in a different department or on different applications.

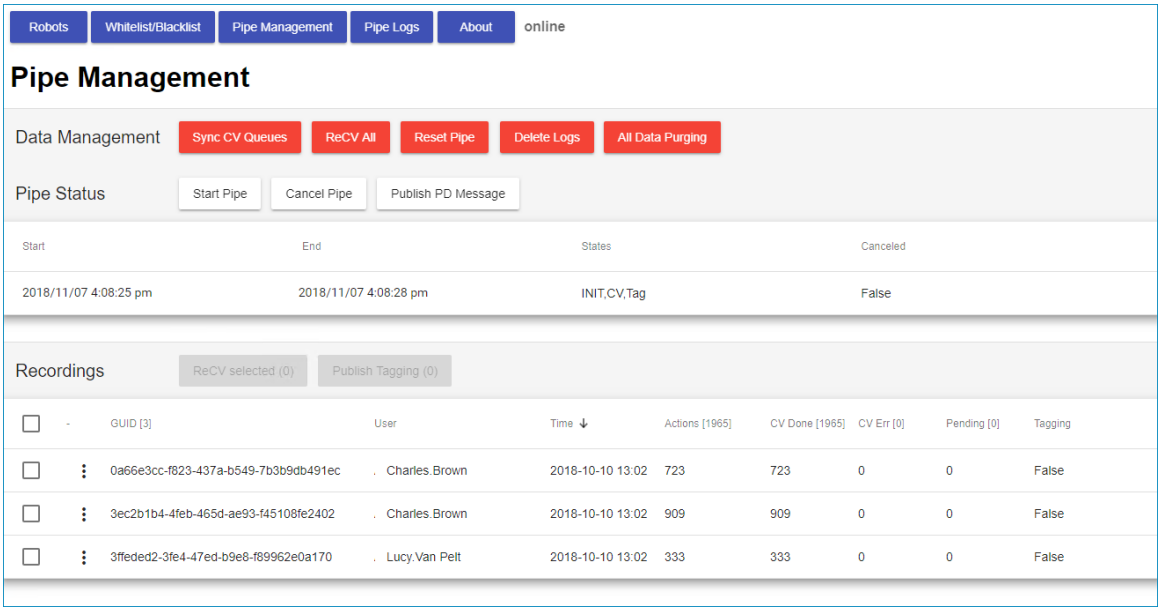
Clearly, this is a procedure that should be undertaken with great caution, and you should be quite sure you have backed up the database and/or extracted all relevant information and/or exported all desired automations before proceeding.

Access Orchestrator with data management options

Access Orchestrator as usual, but make these slight changes to the URL:

- Go straight to the Pipe Management screen by adding `pipe` to the URL; and
- Enable access to the data management options by adding the parameter `?admin=true` to the URL
    - So, the final URL would be: `http://{serverIP}:8788/app/pipe?admin=true`

The **Pipe Management** screen will open with data management options enabled:



Delete data as desired

You have several options at this point for deleting the data you wish to delete.

Deleting data from the entire database

The red buttons allow you to delete data from the entire database:

Reset Pipe — Deletes all tagging data from the database (resets to before the tagging step of the pipe)

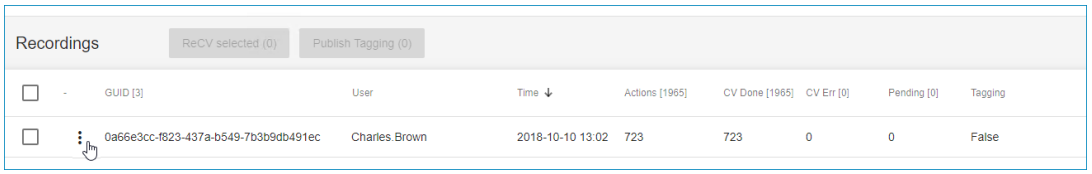
All Data Purging — Deletes all data from the database (only do this if you are very, very sure this is what you want to do)

After you click one of these buttons, you will receive a confirmation message, warning that the action is irreversible. If you wish to continue, click the Delete button.

Deleting a specific recording

To delete a specific recording:

1. Click on the menu to the right of the recording you want to work with:



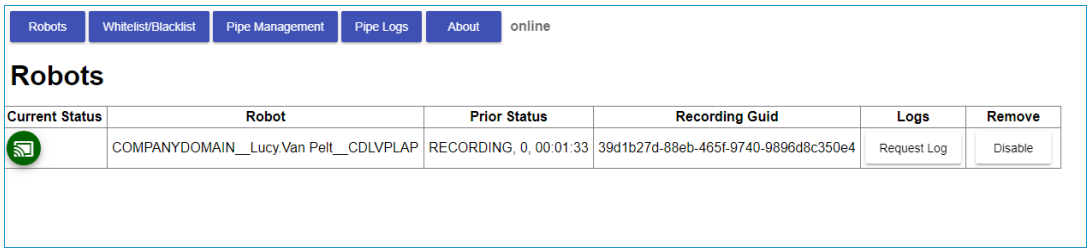
2. From the menu that opens, select **Delete Recording**

Managing Robots

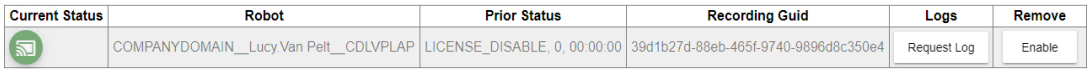
Orchestrator allows you to turn **Discovery Robots** on/off without having to manage them from the client machines themselves.

Turning a Discovery Robot on/off

1. Access Orchestrator
2. Click the Robots tab
3. The following screen will open:



4. To turn a **Discovery Robot** off, click the **Disable** button
5. A robot that has been turned off appears like this:



NOTES

What happens when a Discovery Robot is turned off?

- **How quickly is it turned off?**

  Essentially immediately.

- **Is it still sending data to the server?**

  No.

- **Is data lost?**

  Any data recorded prior to turning the robot off will still be processed. But any activity that occurs on the machine while the robot is turned off will not be recorded.

- **Can the robot be turned back on?**

  Yes, the robot can be turned back on at any time (simply clicking the **Enable** button).

Can a Discovery Robot be turned off when it is not connected to the server?

When a Discovery Robot is not connected to the server (i.e., the icon in the **Current Status** column is red), it can not be turned off. When this is the case, clicking the **Disable** button will have no effect:

- The **Disable** button will not become an **Enable** button
- The robot's row in the table will not turn gray

Wait until the robot is connected to the server (i.e., the icon in the **Current Status** column is green) in order to turn it off.

Managing the Algorithm

The Kryon Process Discovery algorithm is extremely powerful – recording and processing all user actions (in applications and websites that are on the whitelist and/or not on the blacklist). As you use Process Discovery, you may find that you wish to narrow its focus in order to provide more streamlined results and make analysis easier. You can do so by customizing the algorithm's default settings.

TIP

Try... then modify

The default algorithm configuration is the recommended starting point. We recommend initially running Process Discovery with the default settings, then modify them as necessary if you wish to narrow/broaden its results.

Default algorithm configuration

Algorithm settings are managed in the file `C:\Program Files\Kryon Process DiscoveryServer\Pipe\Kryon.ABPD.ProcessDiscovery.exe.config`.

The default configuration is as follows:

```
<!-- Process Discovery params-->
<add key="pdMaxBreakTimeSec" value="1200"/>
<!-- DEFAULT 1200 seconds = 20 minutes -->
<add key="pdProcessMinTimeSec" value="40"/>
<add key="pdProcessMaxTimeSec" value="2400"/>
<!-- DEFAULT 2400 seconds = 40 minutes -->
<add key="pdMinRepeatsForBP" value="5"/>
<add key="MaxRepeatsOfStartStopInside" value="1"/>
<add key="MinShowsPerVariant" value="1"/>
<add key="pdMaxDistBetweenVariants" value="3"/>
<add key="pdMatchProcesesTH" value="90"/>
<!-- DEFAULT 90%. values are [0...100%] -->
<add key="pdMatchVariantsTH" value="95"/>
<!-- DEFAULT 95%. values are [0...100%] -->
<add key="pdPreferLongProcesses" value="true"/>
<add key="PdDebugMode" value="false"/>
<add key="pdMultiStartPoints" value="false"/>
<add key="PdMaskImages" value="false"/>
```

Optional settings

The following settings can be edited in order to narrow/broaden Process Discovery results:

Original key(s)	Description/optional change
`<add key="pdProcessMinTimeSec" value="40"/>` `<add key="pdProcessMaxTimeSec" value="2400"/>`	Set the minimum and maximum duration of a series of actions (in seconds) in order for Process Discovery to consider it a process. The default minimum value of 40 seconds and maximum value of 2400 seconds (40 minutes) can be changed as required: • Lower minimum and/or higher maximum value → more identified processes • Higher minimum and/or lower maximum value → fewer identified processes
`<add key="pdMinRepeatsForBP" value="5"/>`	Sets the minimum number of times a series of user actions must take place in order for Process Discovery to consider it a process. The default value of 5 can be changed as required: • Lower value → more identified processes • Higher value → fewer identified processes
`<add key="pdMaxDistBetweenVariants" value="3"/>`	Determines how similar variants must be in order to be considered a single process. The default value of 3 can be changed as required: • Lower value → more identified processes, each containing fewer variants • Higher value → fewer identified processes, each containing more variants

APPENDIX A: Additional Configuration Options

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

Database & Discovery Server on Different Machines

Where the database is installed on Machine A and the **Discovery Server** is installed on Machine B:

1. Open port **3306** in your network firewall and the Windows firewall
2. From the source location on Machine B, open the following files:
   a. `my.common.ini`
   b. `my.mariadb.ini` (if you installed MariaDB); ***or***
      `my.mysql.ini` (if you installed MySQL)

NOTE

Source location = `{the folder in which the core Discovery Server files were installed}\Support\`

- The default source location is `C:\Program Files\Kryon Process Discovery Server\Support\` (assuming `C` was the local drive with the most free space at the time of installation)
- If you specified a different InstallFolder location during installation, the source location is `{InstallFolder}\Support\`

3. From the destination location on Machine A, open the `my.ini` file

NOTE

Destination location = `{MariaDB_installation_folder}\data\`; or `{MSQL_installation_folder}\data\` (depending on the database program you are using)

4. Merge `my.common.ini` and `mymariadb.ini`/`my.mysql.ini` into `my.ini`

NOTE

What is meant by "merge"?

- Copy to `my.ini` any **lines** that appear in `my.common.ini` or `mymariadb.ini`/`my.mysql.ini` but do not appear in `my.ini`
- Use the **values** from `my.common.ini` and `mymariadb.ini`/`my.mysql.ini` to overwrite the **values** in `my.ini` for lines that appear in both locations
- Leave unchanged any **lines** in `my.ini` that do not appear in `my.common.ini` or `mymariadb.ini`/`my.mysql.ini`

5. From the **Windows Services** app or the **Windows Task Manager > Services** tab on Machine A, restart the **MySQL** service
   - **NOTE:** Yes, even if you are installing MariaDB, the name of the service that needs to be restarted is MySQL

Non-Default Port for Communication Between Discovery Robots & Discovery Server

NOTE

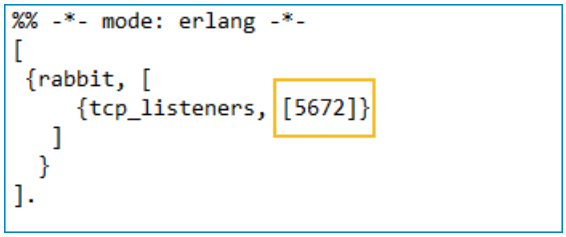
This section is relevant only when installing without TLS. (For TLS installations, follow the instructions in Appendix B.)

By default, the communication between **Discovery Robots** and the **Discovery Server** uses port `5672`. To use a port other than `5672`, follow these instructions:

On the Discovery Server

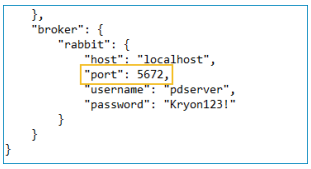
Edit the rabbitmq.config file

1. Right-click the **Windows Start** button
2. Select **Run**
3. Type `%appdata%/RabbitMQ`, then hit <ENTER>
4. From the window that opens, open the `rabbitmq.config` file with a text editor
5. Change the value of `tcp_listeners` from `5672` to the port number you are using



Edit Product Discovery configuration files

1. With a text editor, open each of the configuration files listed in the table below, and change the value of the specified key from 5672 to the port number you are using
   - **NOTE:** In the following table, `{MainPDFolder}` = the folder in which the core **Discovery Server** files were installed
     - By default, this folder is `C:\Program Files\Kryon Process Discovery Server\` (assuming `C` was the local drive with the most free space at the time of installation)
     - If you specified a different InstallFolder location during **Discovery Server** installation, `{MainPDFolder}` is the InstallFolder you specified

Location	File	Key
{MainPDFolder}\Server\Pipe	Kryon.ABPD.PipeRunner.exe.config	<add key="messagesBrokerPort" value="5672"/>
{MainPDFolder}\Server\Pipe	Kryon.ABPD.PipeRunnerService.exe.config	<add key="messagesBrokerPort" value="5672"/>
{MainPDFolder}\Server\Orchestrator\config	production.json	

On Discovery Robots

Edit the Discovery Robot configuration file

1. Open the file {MainRobotFolder}\pddr.exe.config with a text editor and edit the line <add key="messagesBrokerPort" value="5672"/>, and change the value from 5672 to the port number you are using
   - **NOTE:** The {MainRobotFolder} = the folder in which the **Discovery Robot** files were installed
     - By default, this folder is C:\Program Files\PDDR
     - If you specified a different location during **Discovery Robot** installation, the {MainRobotFolder} is the folder you specified

APPENDIX B: TLS Configuration

In this chapter:

Configuring TLS on the Discovery Server

Obtain certificate files

Three certificate files, issues by an authorized CA (certification authority), are required to establish secured TLS communication:

1. CA bundle
2. Server certificate
3. Server private key

These files:

- Must be in PEM or CRT format; and
- Can be installed in a location of your choice on the Discovery Server

TIP

To verify that a certificate is in PEM /CRT format, open it with a text editor. You should see lines that look similar to:

`"-----BEGIN CERTIFICATE-----"`; or

`"-----BEGIN RSA PRIVATE KEY-----"`

Edit the rabbitmq.config file

1. Right-click the **Windows Start** button
2. Select **Run**
3. Type `%appdata%/RabbitMQ`, then hit <ENTER>
4. From the window that opens, open the `rabbitmq.config` file with a text editor
5. Edit the file to read as follows:

```
%% -*- mode: erlang -*-
[
 {ssl, [{versions, ['tlsv1.2']}]},
  {rabbit, [
    {ssl_listeners, [5671]},
      {ssl_options, [{cacertfile, "/path_to/ca_certificate_bundle.pem"},
                     {certfile, "/path_to/certificate.pem"},
                     {keyfile, "/path_to/private_key.pem"},
                     {ciphers, [{rsa,aes_256_gcm,null,sha384}]},
                     {verify, verify_none},
                     {fail_if_no_peer_cert, false} ,
                     {versions, ['tlsv1.2']}]}
  ]}
].
```

NOTES

Provide actual path & certificate file names

In the above file, change the entries beginning with `/path_to/` to the actual paths and filenames of your certificate files. (Note that the paths contain ***forward*** slashes.)

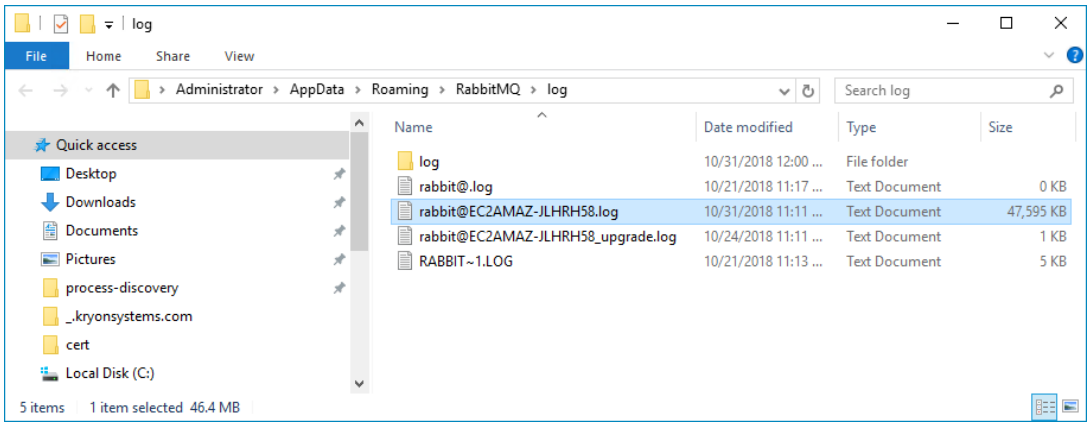
Changing the default port

By default, Process Discovery uses port `5671` for TLS communications. If you wish to use a different port, replace `5671` in the above file with the port of your choice.

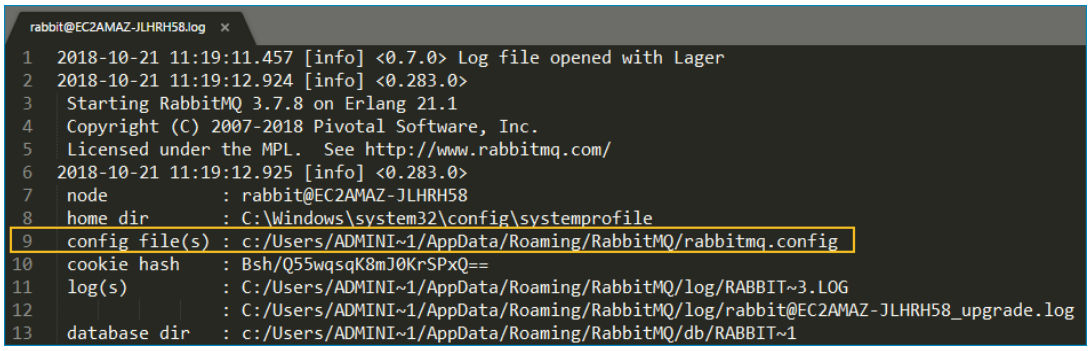
Verify that rabbitmq.config is loading properly

1. Right-click the **Windows Start** button
2. Select **Run**
3. Type `%appdata%/RabbitMQ/log`, then hit <ENTER>
4. From the window that opens, open the largest log file with a text editor

   RECOMMENDED: Use a text editor other than Windows Notepad (such as Notepad++ or Sublime Text). Notepad eliminates line breaks when opening the file, making it almost impossible to read!



5. Confirm that **rabbitmq.config** is the configuration file being loaded:



6. If you see that the configuration file is not being loaded properly, follow these troubleshooting steps:

TROUBLESHOOTING

Repairing an incorrectly loading configuration file

a. Ensure that the folder **%AppData%/RabbitMQ** contains only one configuration file: **rabbitmq.config**
   - If there is a file called **rabbitmq.conf**, delete it
   - If there is a file called **advanced.config**, ensure that it is empty (it should contain only `[]`)

b. Run the CMD prompt as an administrator, and change the directory to `{rabbit_installation_folder}\sbin`
   - The rabbit installation folder is generally `C:\Program Files\RabbitMQ Server\rabbitmq_server-3.7.4` (or something similar)

c. Remove the rabbitmq service by typing the following command:

```
rabbitmq-service.bat remove
```

d. Reinstall the service by typing:

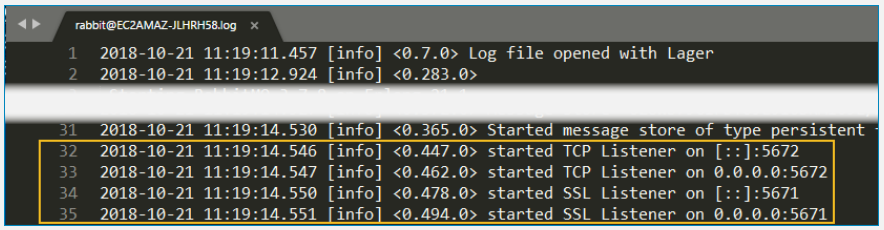
```
rabbitmq-service.bat install
```

e. Start the service by typing:

```
rabbimq-service.bat start
```

f. Return to step 1 above to open the log file and verify that `rabbitmq.config` is now loading properly

g. Check a bit further down in the log file and ensure that the service is listening on 2 ports: regular and secured:



Configuring TLS on Discovery Robots

To configure TLS on **Discovery Robots**, open the file `{MainRobotFolder}\pddr.exe.config` with a text editor and edit the following keys as detailed below.

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

Original parameter	Required change
`<add key="TlsEnabled" value="false"/>`	`<add key="TlsEnabled" value="`**`true`**`"/>`
`<add key="TlsServer" value=""/>`	`<add key="TlsServer" value="`**`{certificate_server_name}`**`"/>` • `{certificate_server_name}` in the key above must be the name exactly as specified on the server certificate file
`<add key="TlsPort" value="5671"/>`	`<add key="TlsPort" value="`**`{ssl_listeners_port`**`"/>` • Change this key only if you are using a port other `5671` for TLS communications • This should be the same port as you set for `ssl_listeners` (in the `rabbitmq.config` file) when configuring the Discovery Server

Troubleshooting TLS Configuration

Follow these basic steps to troubleshoot connectivity issues with a TLS configuration:

Eliminate the possibility of firewall issue

- Confirm that ports 5672 (regular port) and 5671 (secured port) are open both on your network firewall and the Windows firewall

Eliminate the possibility of a TLS certificate issue

- Install a Discovery Robot on the same machine as the Discovery Server
- If there is connectivity, you have isolated a network issue (i.e., you know that there is not a issue with TLS certificates)

Troubleshoot a network issue

- Install and open Wireshark on the server side, and check to see if the `client hello` message of the TLS handshake has arrived
- If it has not, install and open Wireshark on the client side, and check to see if the `client hello` message has been sent
  - If the message has been sent but the server doesn't answer, we know that there is a server-side issue
  - If the message has not been sent, we know that there is a client-side issue
  - We know that the client is connected to server if we see a `client hello` message on the server side

APPENDIX C: Enhanced Security Options for Discovery Console

You can elect to add any or all of the following **custom HTTP response headers** to enhance the security of **Discovery Console**, in accordance with your organization's needs:

1. X-XSS-Protection
2. Strict-Transport-Security
3. X-Content-Type
4. X-Frame-Options
5. Cache-Control
6. Pragma
7. Access-Control-Allow-Origin

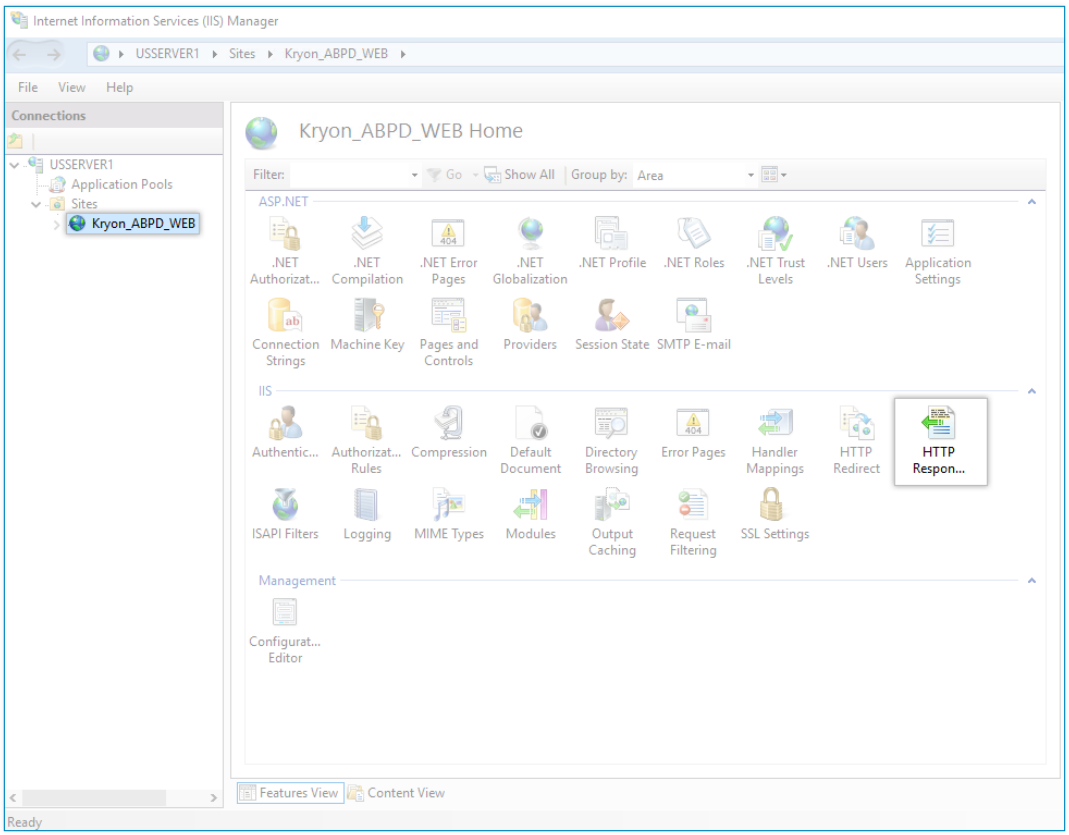
Each of the response headers is:

- Configured through **Internet Information Services (IIS) Manager** from the machine on which **Discovery Console** is installed; and
- Applicable to the **Kryon_ABPD_WEB** site (that was automatically configured in IIS when the **Discovery Server** was installed)

Configuring custom HTTP response headers

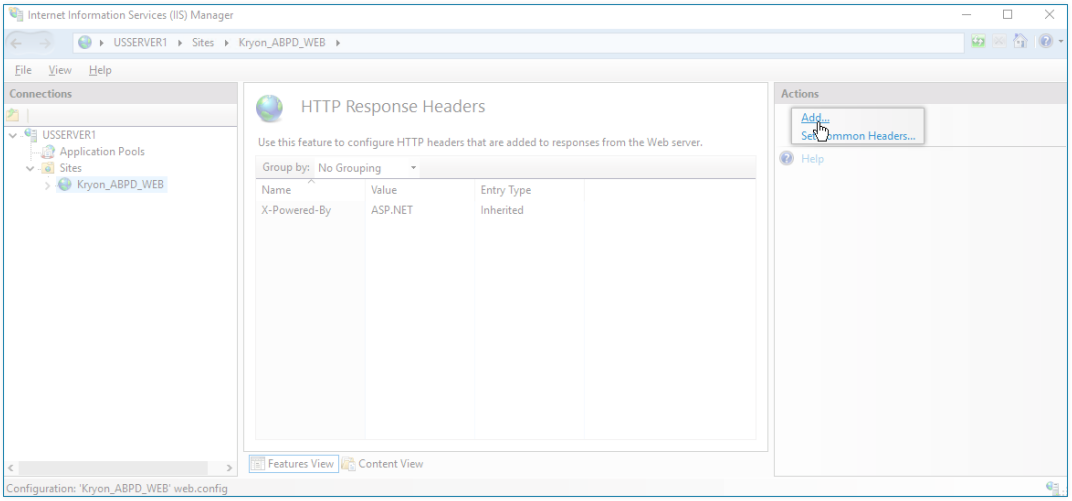
To add custom HTTP response headers, follow these steps:

1. In the **Connections** panel, click on the **Kryon_ABPD_WEB** site
2. Double-click on the **HTTP Response Headers** icon

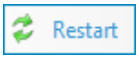


The **HTTP Response Headers** screen will open

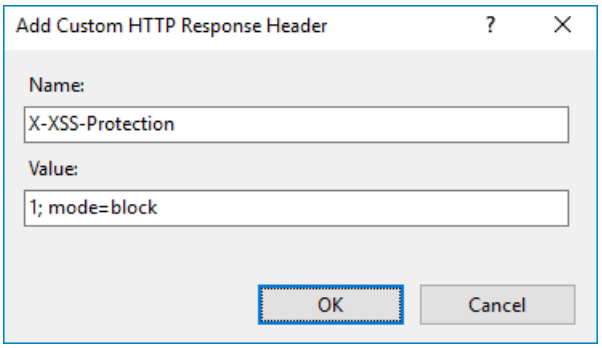
3. In the **Actions** panel, click **Add**



The **Add Custom HTTP Response Header** dialog box will open

4. Enter the header **Name** and **Value** as indicated in the following sections
5. Click **OK** to save the response header
6. Repeat these steps for each response header you wish to add
7. When you are finished adding response headers, restart the IIS server:
   a. In the **Connections** panel, click on the server name
   b. In the **Actions** panel, click 

X-XSS-Protection

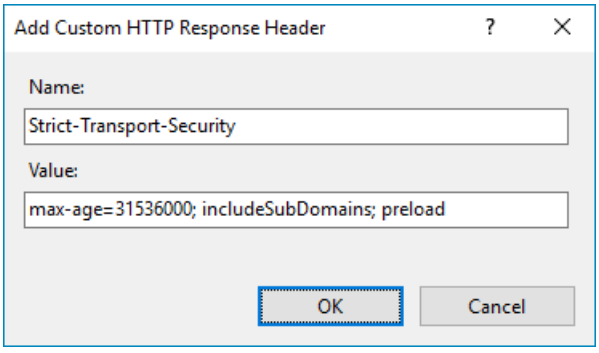


The **X-XSS-Protection** HTTP response header is a feature of Internet Explorer, Chrome and Safari that stops pages from loading when they detect reflected cross-site scripting (XSS) attacks.

Name: `X-XSS-Protection`

Value: `1; mode=block`

Strict-Transport-Security

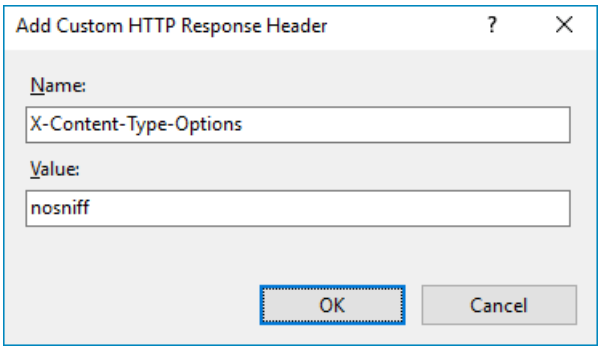


The **Strict-Transport-Security** HTTP response header (often abbreviated as HSTS) allows a website instruct browsers that it should only be accessed using HTTPS, instead of using HTTP.

Name: `Strict-Transport-Security`

Value: `max-age=31536000; includeSubDomains; preload`

X-Content-Type

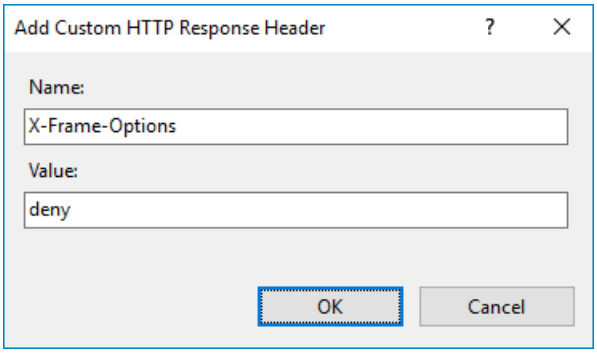


The **X-Content-Type-Options** HTTP response header is a marker used by the server to indicate that the MIME types advertised in the Content-Type headers should not be changed.

Name: `X-Content-Type-Options`

Value: `nosniff`

X-Frame-Options

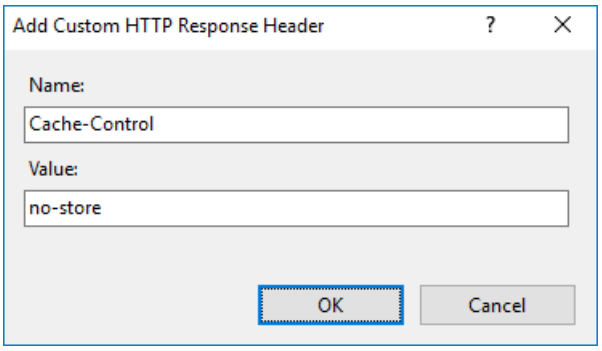


The **X-Frame-Options** HTTP response header can be used to indicate whether or not a browser should be allowed to render a page in a `<frame>`, `<iframe>` or `<object>`. It is used to avoid clickjacking attacks, by ensuring that site content is not embedded into other sites.

Name: `X-Frame-Options`

Value: `deny`

Cache-Control

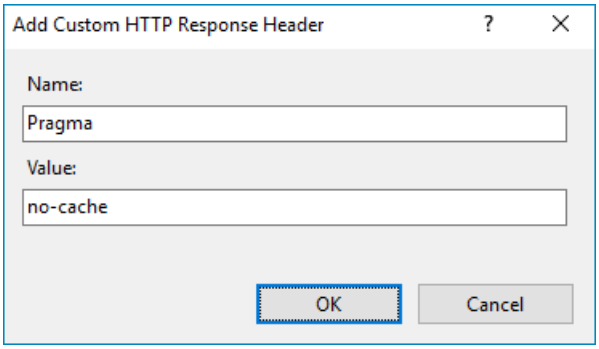


The **Cache-Control** HTTP response header is used to specify directives for caching mechanisms in both requests and responses. (The no-store value directs that the cache should not store anything about the client request or server response.)

Name: `Cache-Control`

Value: `no-store`

Pragma

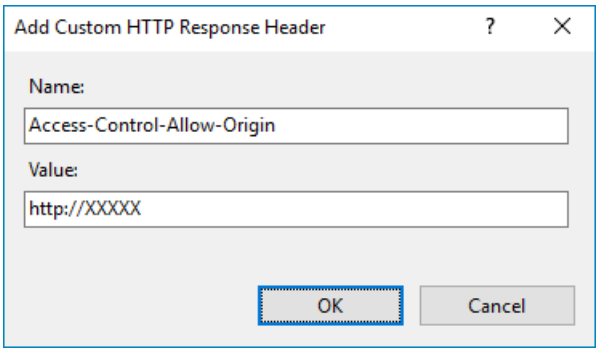


The **Pragma** HTTP response header is an implementation-specific header that is used for backwards compatibility with HTTP/1.0 caches where the Cache-Control HTTP/1.1 header is not yet present.

Name: `Pragma`

Value: `no-cache`

Access-Control-Allow-Origin



The **Access-Control-Allow-Origin** HTTP response header indicates whether the response can be shared with requesting code from the given origin.

Name: `Access-Control-Allow-Origin`

Value: `http://XXXXX` (where XXXXX is the name, IP address, or domain of the **Discovery Console** server