

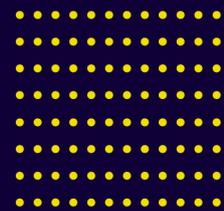


KRYON™

Release Notes

Kryon Process Discovery

V. 21.6



Contents

Kryon Process Discovery 21.6

- New Features and Enhancements 3
 - Image Masking Scheduler 3
 - Cancel or Delete Discovery Results 3
 - Explore Process Variants - BETA 3
- Bug Fixes 4
- Known Issues 6

Technical Details

- Image Masking Scheduler 8
- Canceling or deleting Discovery search 10
- Explore Variants - BETA 11
- Bug Fixes Details 15

Kryon Process Discovery 21.6

NEW FEATURES AND ENHANCEMENTS

Image Masking Scheduler

A new masking service that prepares the required images and masks them upfront to improve masking performance significantly while reviewing the process results.

You can now schedule a desired time to mask the images automatically, so it won't interfere with your work.

[Image Masking Scheduler](#)

Cancel or Delete Discovery Results

Easily cancel a discovery search while in progress, or delete it after completion from the Recent tab. Simply hover over the desired search status to cancel or delete a discovery search.

[Canceling or deleting Discovery search](#)

Explore Process Variants - BETA

Introducing the ability to explore process variants.

Each variant represents one or more occurrences that share the same business sub-flows. Click on each variant to display a map route of an occurrence matching the variant.

You can name the variant, and quickly decide whether to add it to the selected process variants or dismiss it.

[Explore Variants - BETA](#)

BUG FIXES

We've fixed the following bugs:

Item#	Bug	More details
2973	Unexpected error when the Robot attempts to connect over TLS [In Process Discovery Robot]	Although the error does allow the PD Robot to connect and record actions, it eliminates TLS configuration on the server side. more details
2978	PD Robot unable to record FireFox actions in German display	-
3022	Discovery results page mistakenly indicate an "in progress" state even if there are no actual discovery results to present [In Process Discovery Robot]	-
3038	Navigating to recent discoveries in the PD Dashboard redirects to an irrelevant result [Observed in Process Discovery Console]	-
2966	Different number of discovery results appears on the Discovery results drop down in comparison to the results number on the Dashboard [Observed in Process Discovery Console]	-
2903	"404 page not found" is received when clicking an empty state [Observed in Process Discovery Console]	-
2774	HTTPS protocol isn't supported in Process Discovery installation	-
2694	Discovery results page doesn't refresh automatically when discovery tasks change status.	-

	[Observed in Process Discovery Console	
2906	On the Dashboard, the insight value of TOTAL RECORDING TIME is zeroed (0d:0h:0m).	The total recording time should show the duration of all recordings in days, hours, and minutes.
2731	On the Dashboard, the insight value of TOP USED APPLICATIONS shows a blank for some of the applications names.	

KNOWN ISSUES

We have the following known issues:

Item#	Issue	More details and/or workaround
3025	Snipping a process doesn't affect or change the process variants. [Observed in Process Discovery Console]	When snipping a process manually out of the original user recording and saving it to the library, the variants feature for snipped processes isn't supported at this stage.
2240	Inaccurate status indication of unsuccessful process discovery [Observed in Process Discovery Console]	In case of an error during a triggered process discovery search, the search might be stuck "in progress" status. Workaround: Cancel the search manually by clicking the new ' cancel ' button and initiate a new search.
3120	Unable to change the group name after adding it to the variant list. Process Discovery Console]	When adding a variant to the variants list and then attempting to change the group name in the variants map, the name doesn't change.
3050	The error message "Something went wrong - Cannot get variants data, please try again" might redundantly pop-up a second time. Process Discovery Console]	Ignore the second time the message appears.
2772	The end-time stamp of a recorded session might show incorrect values.	The columns Session start time and Session end time under Discovery > recorded sessions might show non-logical values, such as, end-time with a time stamp earlier than the start-time.

Technical Details

For detailed usage explanations, see the following sections:

Image Masking Scheduler	8
Canceling or deleting Discovery search	10
Explore Variants - BETA	11
Bug Fixes Details	15

Image Masking Scheduler

We elevated the **Image Masking** feature to the next level and added to it a new functionality. Now you can schedule **Image Masking** 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.



NOTE

Image Masking has to be enabled and set before you configure the scheduler. Follow the instructions in the [Process Discovery User Guide](#) or [Installation and Upgrade Guide > Image Masking](#).

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:

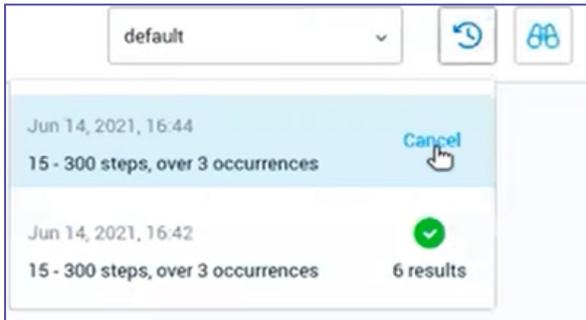
- a. The discovery runs at the scheduled day and time
- b. The results record is saved under recent discoveries (as any other discovery search)
- c. The data collected as part of the scheduled discovery goes through a masking process

Canceling or deleting Discovery search

Cancel an irrelevant discovery search while in progress, or delete it after completion from the Recent searches tab. A quick hover on the desired search status will allow you to cancel or delete the discovery search.

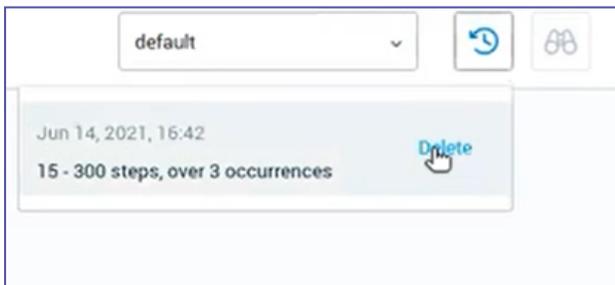
To cancel a running discovery:

Go to **Discovery page / discovery results** > click the **Recent** icon > pick a search result of a discovery **In progress** > click **Cancel**



To delete a completed/no-results discovery:

Go to **Discovery page / discovery results** > click the **Recent** icon > pick a search result of a finished discovery > click **Delete**



Explore Variants - BETA

To understand how the "**Explore Variants**" feature works, let's break down all the components of a discovered process and understand the relationship between them.

Process occurrence :

A discovered process is a process that was performed repetitively by users. Each time a process is identified, it is identified alongside with all its occurrences. I.e., Process occurrences are all the times a process has been performed by user(s).

Process Group :

A Group is a set of identical/very similar actions that deviate from the regular process and appear on 1 or more occurrences, but not on all occurrences.

Process Variant

A Variant is consisted of 1 or more process occurrence(s) that share very similar **Group(s)** of actions.

Lets take the following scenario:

John and Sarah perform the same exact cash order process ~20 times a day. Process Discovery identifies the users' actions as a worthy-process to automate and creates all the different Process Occurrences for the Process Discovery administrator to review on the Process Discovery Console.

Process Discovery algorithm has discovered additional insights:

Each time John and Sarah processed a cash order for an amount > 2k, both of them also sent an approval invoice email to their manager.

Meaning, in some cases, both John and Sarah can deviate from the regular process and performed a *unique and consecutive* set of actions while processing the cash-order.

Example:

Occurrence #	Groups of actions	Variants
A	<div data-bbox="389 383 1243 443" style="background-color: #ffe4c4; border: 1px solid black; padding: 2px;">Calculate the amount of money to cash</div> <div data-bbox="389 443 1243 504" style="background-color: #90ee90; border: 1px solid black; padding: 2px;">Sent an approval invoice email</div>	<div data-bbox="1267 398 1391 488" style="background-color: #add8e6; border: 1px solid black; padding: 2px;">Variant 1</div>
B	<div data-bbox="389 600 1243 660" style="background-color: #90ee90; border: 1px solid black; padding: 2px;">Sent an approval invoice email</div>	<div data-bbox="1267 586 1391 676" style="background-color: #d2b48c; border: 1px solid black; padding: 2px;">Variant 2</div>
C	<div data-bbox="389 772 1243 833" style="background-color: #ffe4c4; border: 1px solid black; padding: 2px;">Calculate the amount of money to cash</div> <div data-bbox="389 833 1243 893" style="background-color: #90ee90; border: 1px solid black; padding: 2px;">Sent an approval invoice email</div>	<div data-bbox="1267 788 1391 878" style="background-color: #add8e6; border: 1px solid black; padding: 2px;">Variant 1</div>
D	<div data-bbox="389 967 1243 1028" style="background-color: #ffe4c4; border: 1px solid black; padding: 2px;">Calculate the amount of money to cash</div> <div data-bbox="389 1028 1243 1088" style="background-color: #dda0dd; border: 1px solid black; padding: 2px;">Scanned invoice to Salesforce</div>	<div data-bbox="1267 976 1391 1066" style="background-color: #add8e6; border: 1px solid black; padding: 2px;">Variant 3</div>

So how are the variants formatted based on the above 4 occurrences?

Occurrence A and C are identical as they are consisted of the same groups of actions. This means they share the same variant = Variant 1

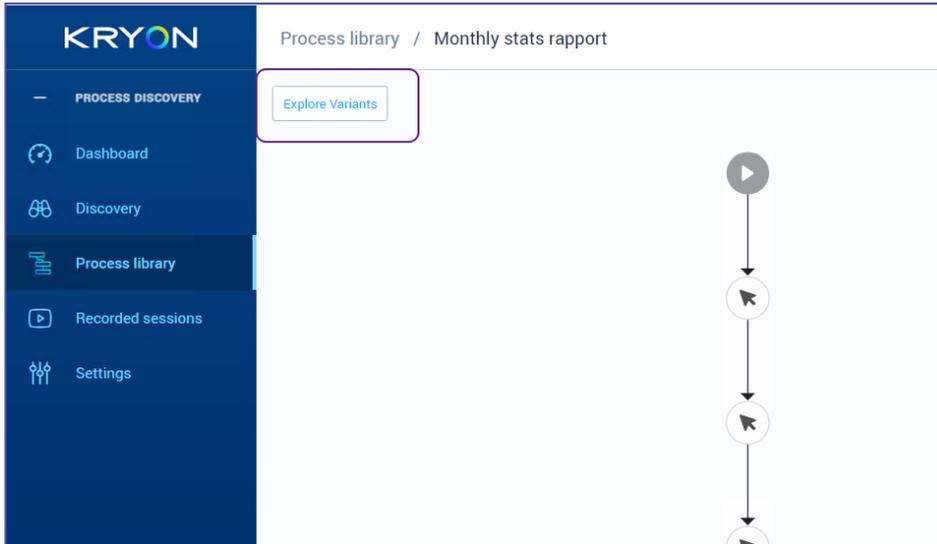
Occurrence B is unique = Variant 2

Occurrence D is unique = Variant 3

So, what's the system behavior in such scenario?

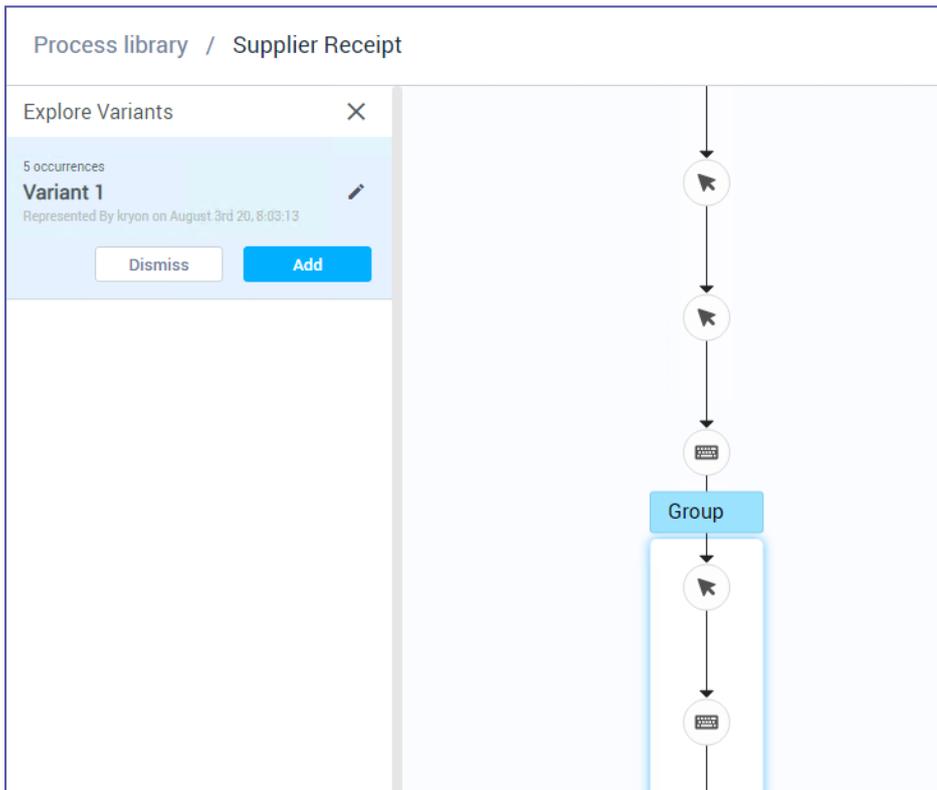
The system presents the identified Groups that are worth exploring.

Where? Process Library page > select a process from the list to review its potential variants > Click **Explore Variants** if available.



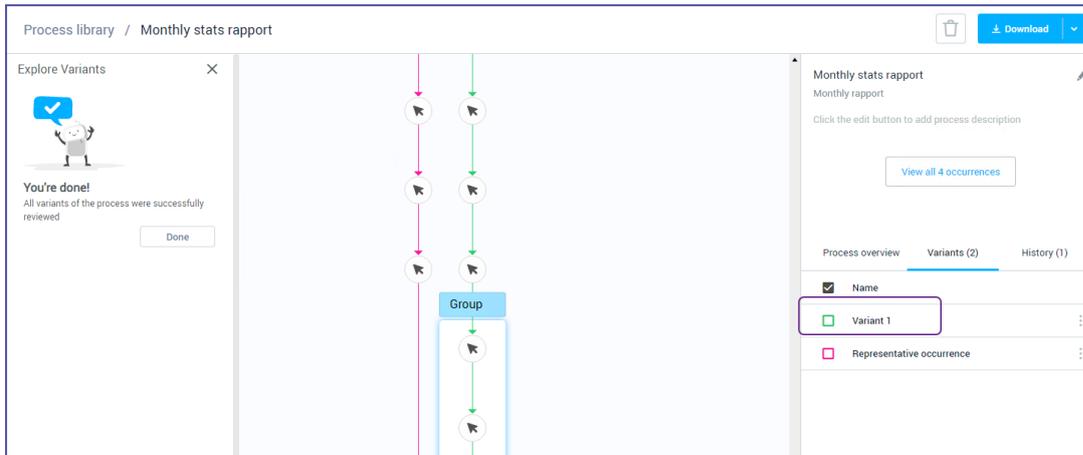
What happens when you click Explore Variants?

Once you click **Explore Variants**, the system opens for you a list of variants to explore (Variant 1, Variant 2, Variant 3, etc.). Once you click on a variant name, it takes you to the point in time where it identified a unique consecutive **Group** of actions in the process occurrence.

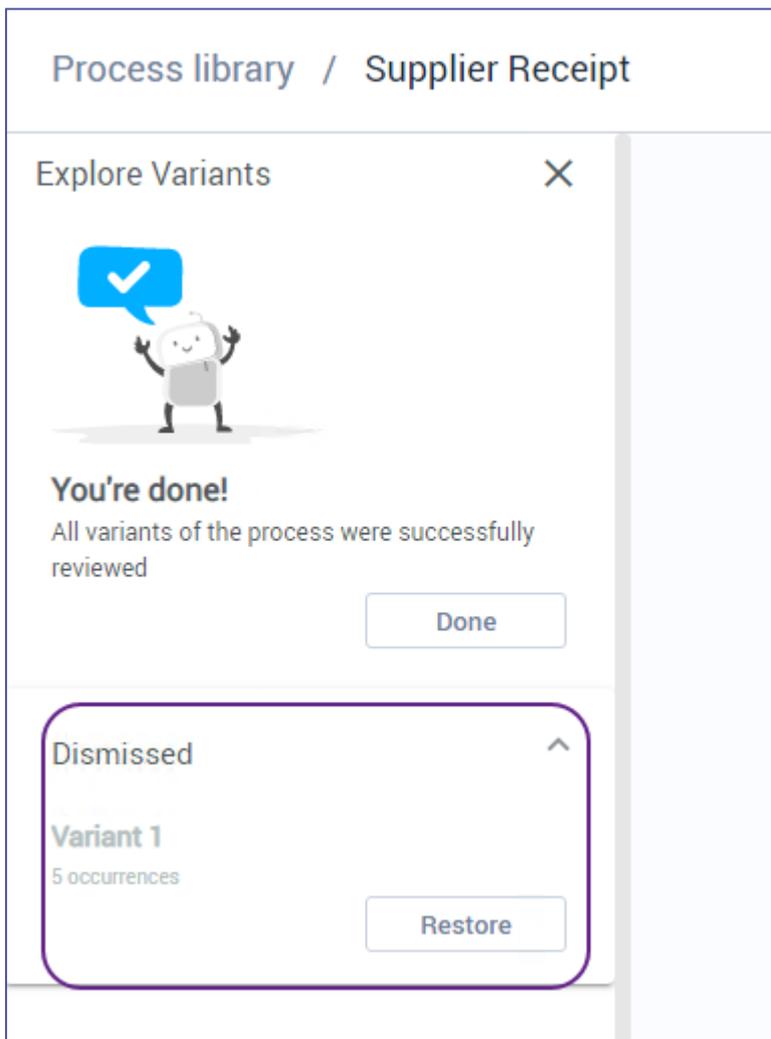


At this point, all you need to do is to review the Group steps to decide whether to add the variant to the list of approved variants or to dismiss it:

If added, the variant will appear on the list on the right after finishing and exiting the "explore variants" mode:



If dismissed, you can always come back to it and restore it:



Bug Fixes Details

Robot unable to connect to server over TLS configuration

The following error is received once the Robot attempts to connect to the server over TLS:

```
{
  "unit": "pddr",
  "uniqueIdentifier": "PDDR-19CECD9CAE0EDB1FA38299B9D34E66B55454926C966A0657DA3B741247F2D5CC",
  "level": "Error",
  "time": 1623864500,
  "message": "Failed while sending a message.
  Error -> System.Net.Http.HttpRequestException: An error occurred while sending the request.
  ---> System.Net.WebException: The request was aborted: Could not create SSL/TLS secure channel.
  \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
  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 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__  
32.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__  
27.MoveNext()"}
```