

Contents

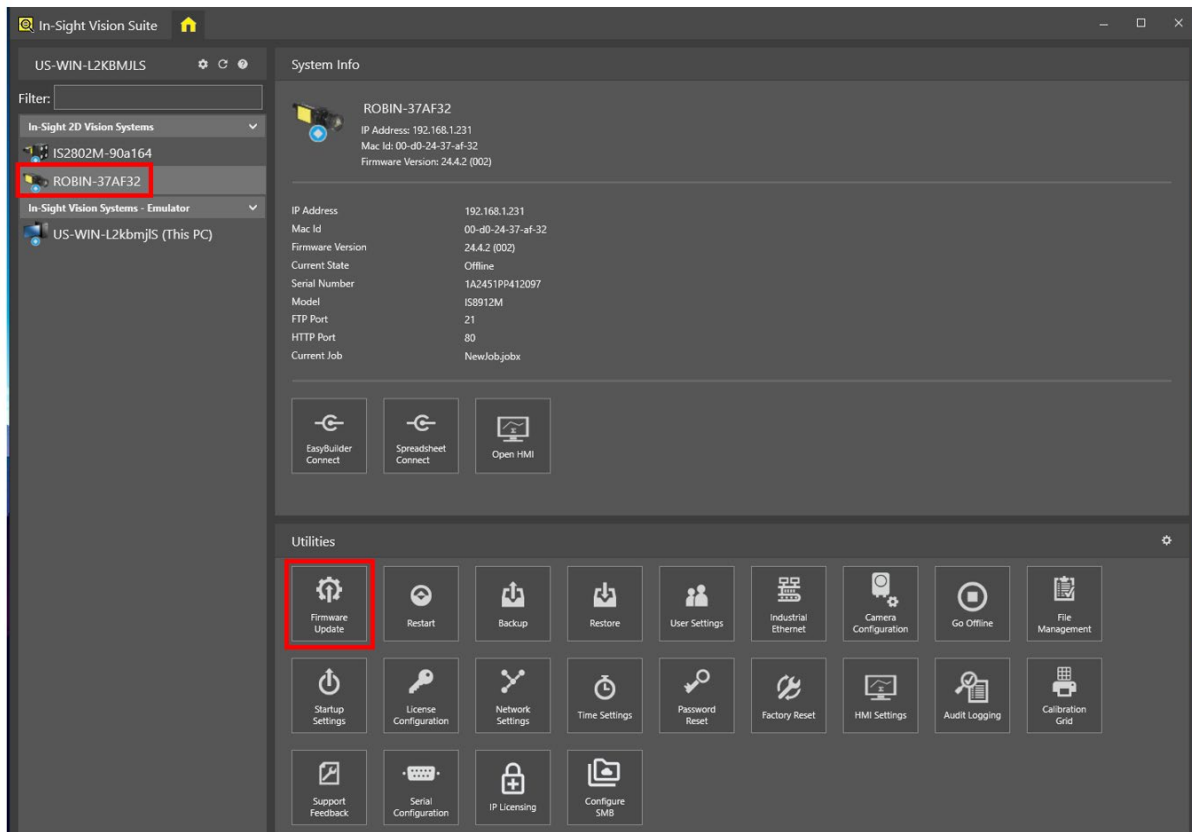
- 1. Setting Up Job Validation..... 2**
 - 1.1 Update Firmware..... 2**
 - 1.2 Image Upload 3**
 - 1.3 Set Up Job Conditions 4**
 - 1.4 Create Validation Set..... 6**
 - 1.5 Run Job Validation 12**
 - 1.6 System Validation..... 14**
 - 1.7 User Control 16**

1. Setting Up Job Validation

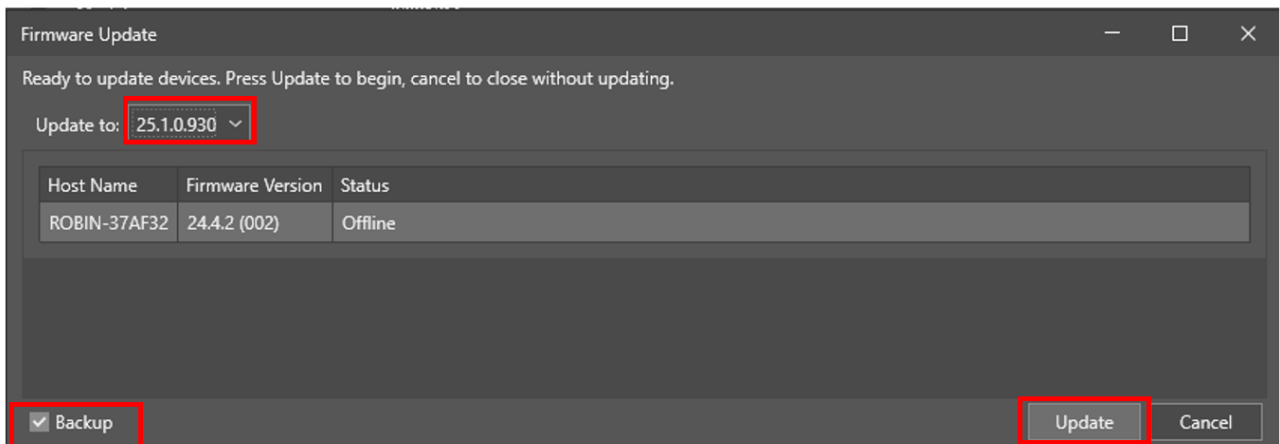
This document serves as a guide to use the In-Sight Vision Suite (ISVS) Job Validation feature in your camera, included in the 25.1.0 release firmware. This feature is intended to help aid in obtaining 21 CFR Part 11 compliance and help ensure the validity of a system's performance throughout its service.

1.1 Update Firmware

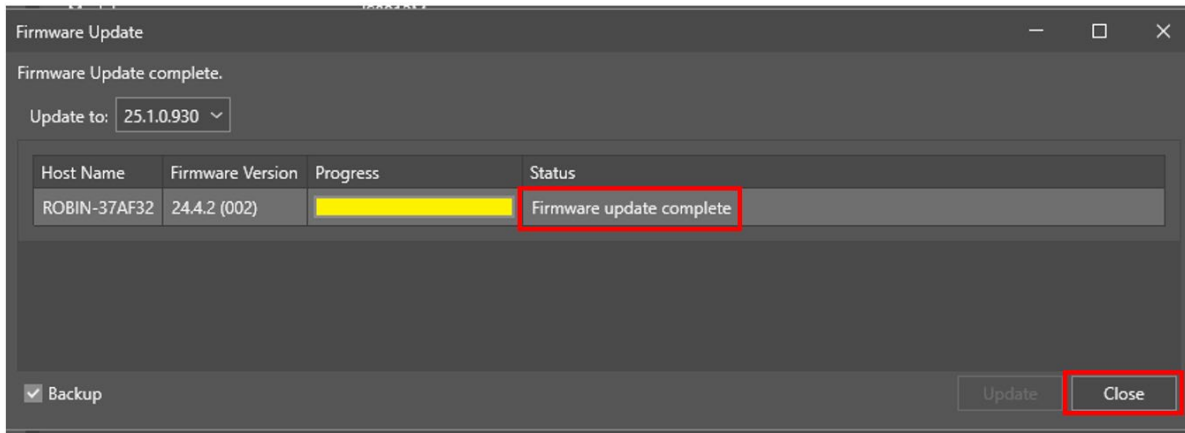
To enable this feature, first update the firmware on the camera to use the 25.1.0 (or newer) version. To do this, install ISVS 25.1.0 (or newer) and run the application. Once in, find your desired camera and click **Firmware Update**.



Take a backup by checking the Backup box.



From here, the system will begin the update process. Once done, you should see a **Firmware update complete** if successful and you can close the window.

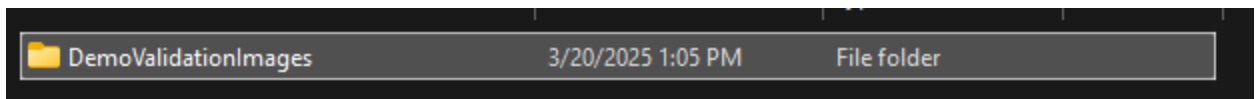


Once updated, connect to your camera and start your job creation within spreadsheet.

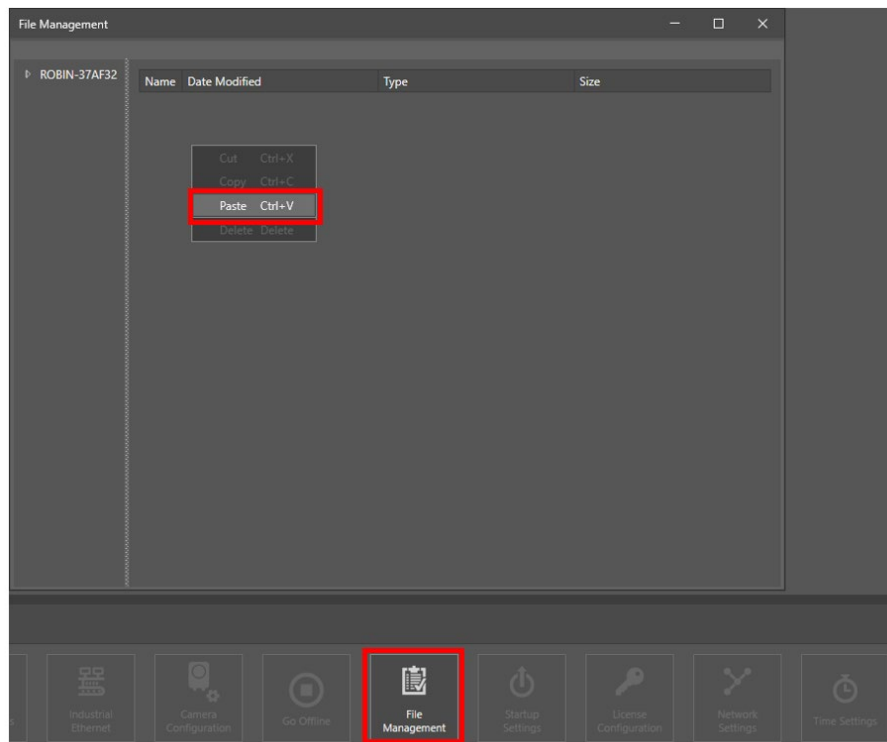
1.2 Image Upload

From here, it's a good idea to upload your images used for validation to the camera and to name/label your images "good" or "bad" beforehand to help keep track of what should pass and what should fail.

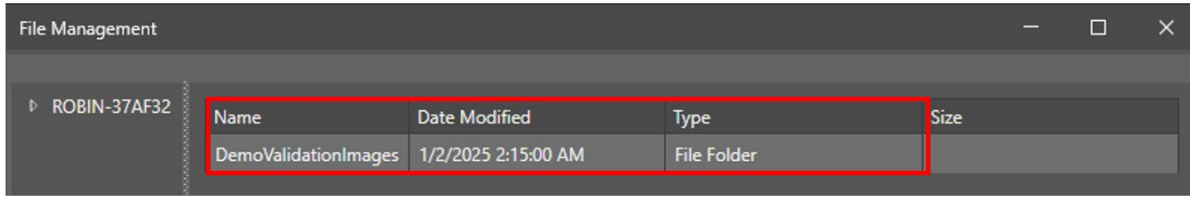
Once you have your images named appropriately, place them in a folder labeled something meaningful.



Copy this folder and paste it into your camera using the File Management utility in ISVS.



Once complete, you should see your image folder copied over.



1.3 Set Up Job Conditions

In this example, we will set up a simple blob tool and count the number of pieces that fall within a certain range of pixel area. For our passing criteria, we will look at allowing a range of areas and a min/max count of blobs that fall within that range by using the **EditInt** function. Once we have our adjustable values, we can reference them when using a blob tool.

User Criteria	
Min Blob Area	50000
Max Blob Area	70000
Min Count	2
Max Count	5

We'll use the **DetectBlobs** tool in conjunction with the **SortBlobs** tool to find up to 10 blobs.

Inspection Tool Test												
Find Blobs												
oBlobs	Index	X	Y	Angle	Color	Score	Area	Elongation	Holes	Perimeter	Spread	
oBlobs		0.000	2328.776	1803.386	84.489	0.000	100.000	57512.000	0.015	0.000	1249.000	0.205
		1.000	1991.884	2156.531	129.749	0.000	100.000	60620.000	0.013	0.000	1305.000	0.200
		2.000	2670.834	2261.788	-6.257	0.000	100.000	64545.000	0.008	0.000	1593.000	0.187
		3.000	2186.107	2635.987	-1.492	0.000	100.000	64737.000	0.008	0.000	1979.000	0.185
		4.000	#ERR	#ERR	#ERR	#ERR	0.000	#ERR	#ERR	#ERR	#ERR	#ERR
		5.000	#ERR	#ERR	#ERR	#ERR	0.000	#ERR	#ERR	#ERR	#ERR	#ERR
		6.000	#ERR	#ERR	#ERR	#ERR	0.000	#ERR	#ERR	#ERR	#ERR	#ERR
		7.000	#ERR	#ERR	#ERR	#ERR	0.000	#ERR	#ERR	#ERR	#ERR	#ERR
		8.000	#ERR	#ERR	#ERR	#ERR	0.000	#ERR	#ERR	#ERR	#ERR	#ERR
		9.000	#ERR	#ERR	#ERR	#ERR	0.000	#ERR	#ERR	#ERR	#ERR	#ERR

Once here, we can add on an **ErrFree** column to check the Area measurements and then create a column for using that value with an **InRange** to check if they fall within our criteria.

Inspection Tool Test													
Find Blobs													
oBlobs	Index	X	Y	Angle	Color	Score	Area	Elongation	Holes	Perimeter	Spread	ErrFree	In Range
oBlobs		0.000	2328.776	1803.386	84.489	0.000	100.000	57512.000	0.015	0.000	1249.000	0.205	1.000
		1.000	1991.884	2156.531	129.749	0.000	100.000	60620.000	0.013	0.000	1305.000	0.200	1.000
		2.000	2670.834	2261.788	-6.257	0.000	100.000	64545.000	0.008	0.000	1593.000	0.187	1.000
		3.000	2186.107	2635.987	-1.492	0.000	100.000	64737.000	0.008	0.000	1979.000	0.185	1.000
		4.000	#ERR	#ERR	#ERR	#ERR	0.000	#ERR	#ERR	#ERR	#ERR	0.000	0.000
		5.000	#ERR	#ERR	#ERR	#ERR	0.000	#ERR	#ERR	#ERR	#ERR	0.000	0.000
		6.000	#ERR	#ERR	#ERR	#ERR	0.000	#ERR	#ERR	#ERR	#ERR	0.000	0.000
		7.000	#ERR	#ERR	#ERR	#ERR	0.000	#ERR	#ERR	#ERR	#ERR	0.000	0.000
		8.000	#ERR	#ERR	#ERR	#ERR	0.000	#ERR	#ERR	#ERR	#ERR	0.000	0.000
		9.000	#ERR	#ERR	#ERR	#ERR	0.000	#ERR	#ERR	#ERR	#ERR	0.000	0.000

From here, we can sum the cells that are in the InRange column.

ErrFree	In Range
57512.000	1.000
60620.000	1.000
64545.000	1.000
64737.000	1.000
0.000	0.000
0.000	0.000
0.000	0.000
0.000	0.000
0.000	0.000
0.000	0.000
Sum	4.000

Next, we will create a separate portion to check that the count is within the min/max count of passing blobs.

User Criteria	
Min Blob Area	50000
Max Blob Area	70000
Min Count	2
Max Count	5
Results	
Count of Passing	4.000
Final PF	1.000

To enable these values into the Job Validation, we need to setup names for these editable and result variables. To do this, click on each cell and give it a name in the cell formula section. The example below, we will call cell B2 "UserMinArea."

We will do this for the remaining cells that we want to validate.

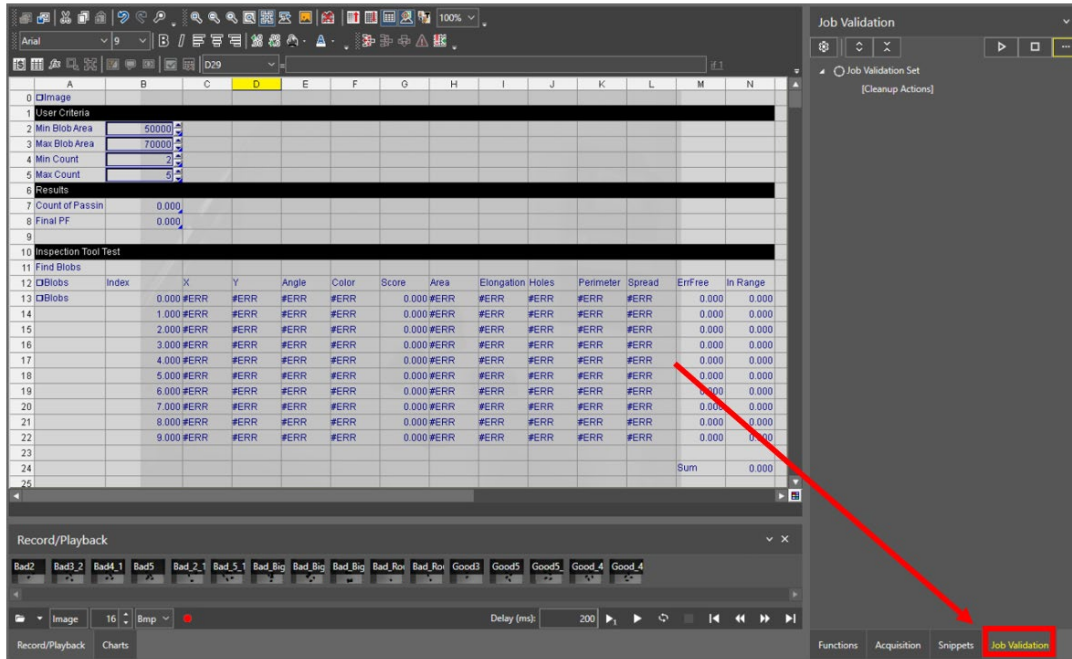
The screenshot shows the Cognex In-Sight Vision Suite software interface. The spreadsheet displays the following data:

	A	B	C	D	E	F	G	H
0	Image							
1	User Criteria							
2	Min Blob Area	50000						
3	Max Blob Area	70000						
4	Min Count	2						
5	Max Count	5						
6	Results							
7	Count of Passing	4.000						
8	Final PF	1.000						
9								

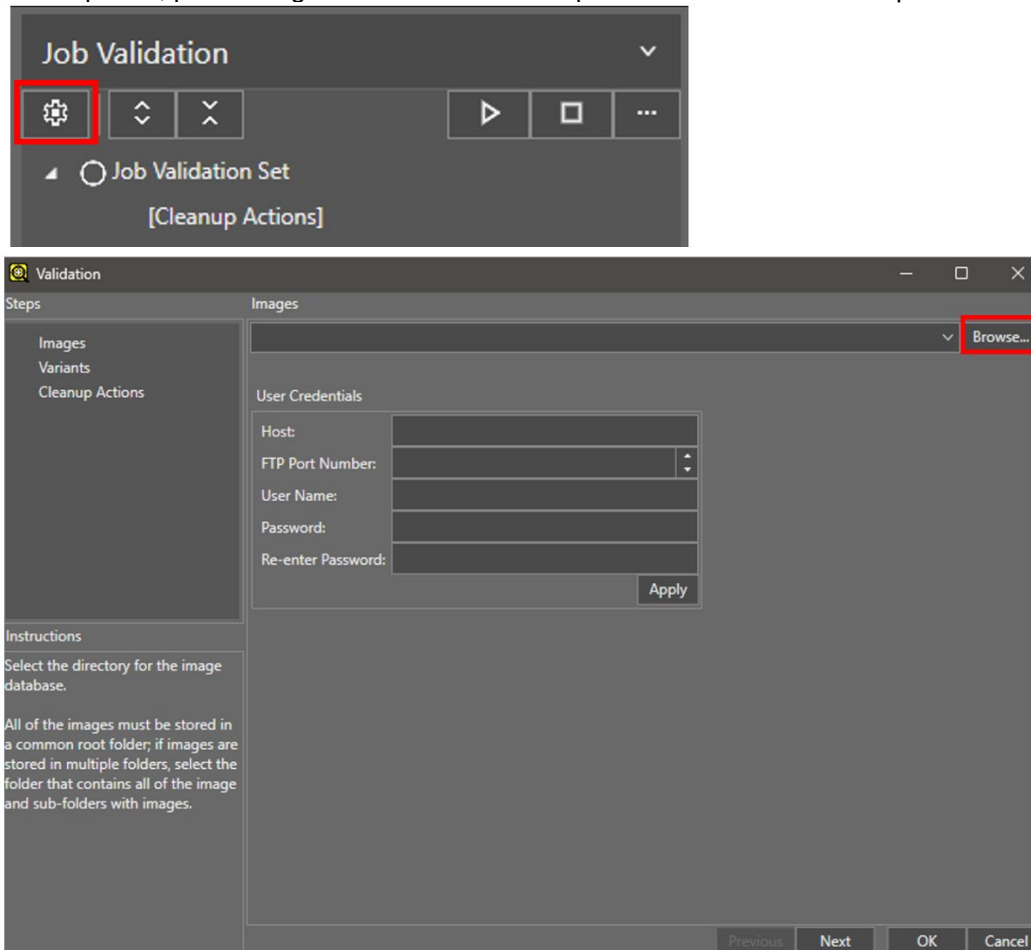
The formula bar at the top shows the formula for cell B2: `EditInt(1,2147483647)`.

1.4 Create Validation Set

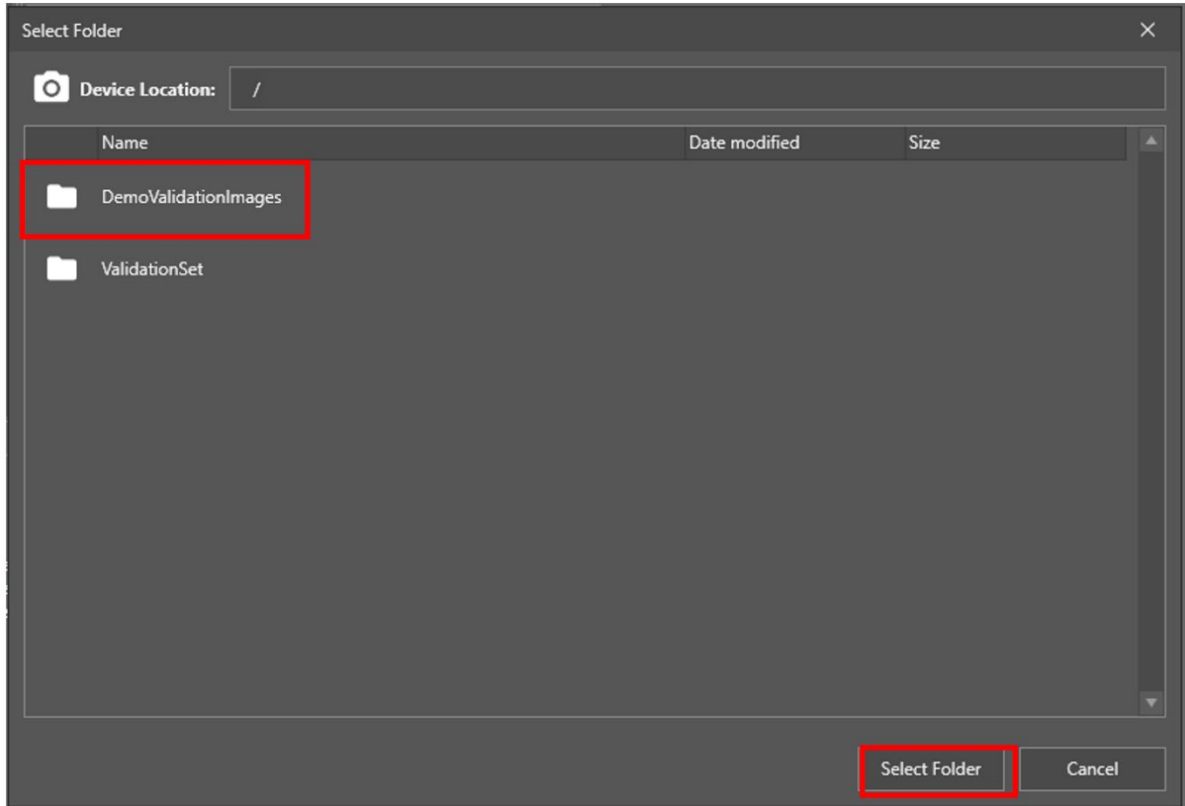
Once your inspection and variables are setup, go to the **Job Validation** section of ISVS.



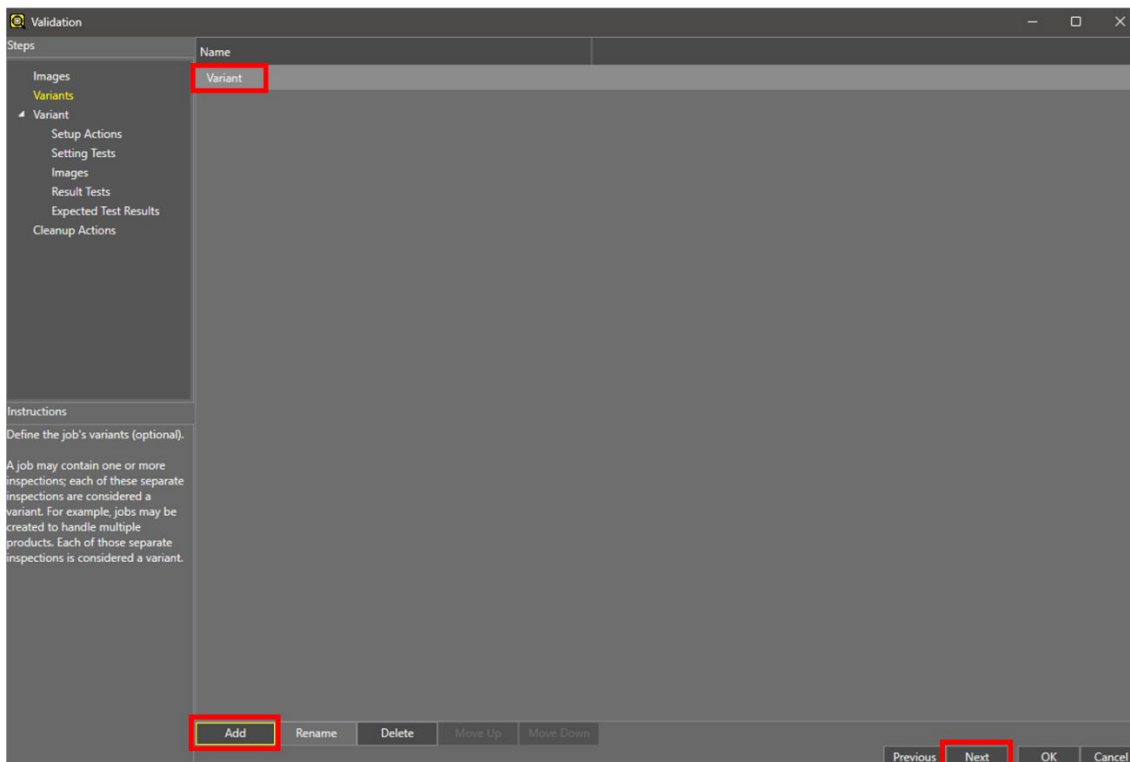
Once opened, press the gear icon towards the top-left to start the validation process.



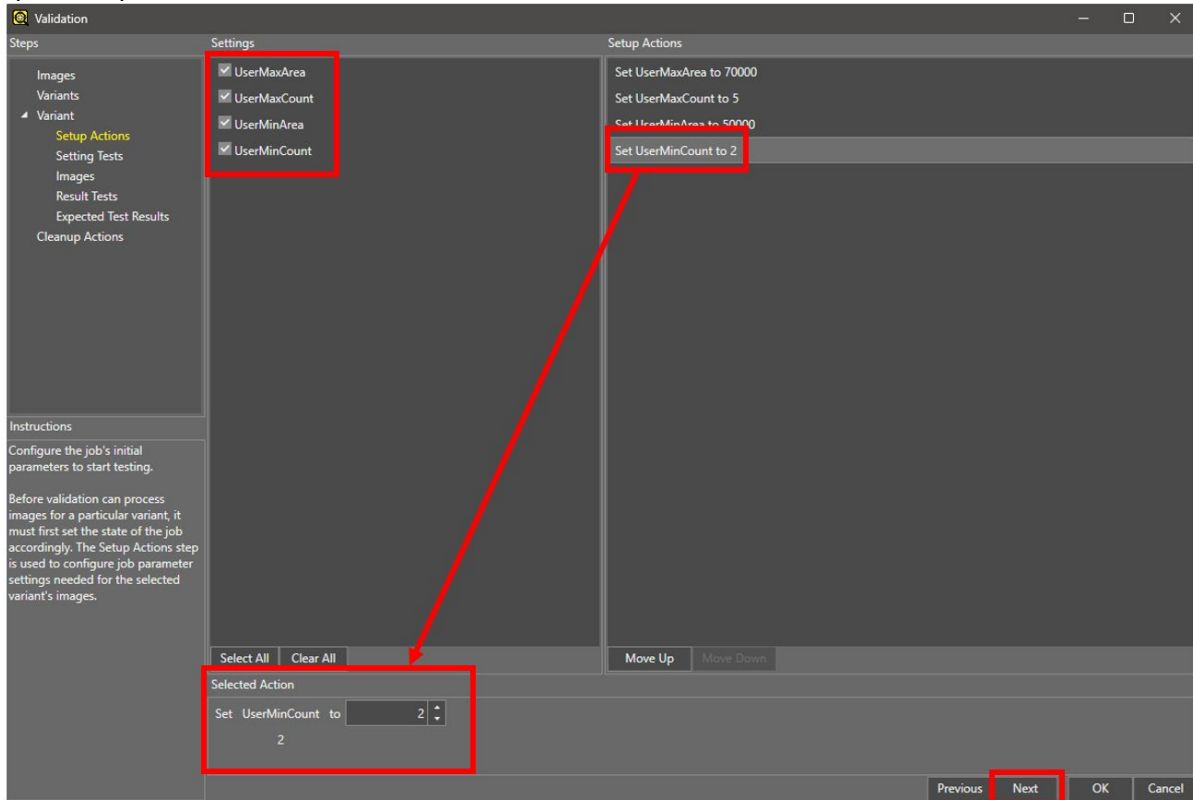
Taking the images that we set up earlier on the camera, we can reference them now.



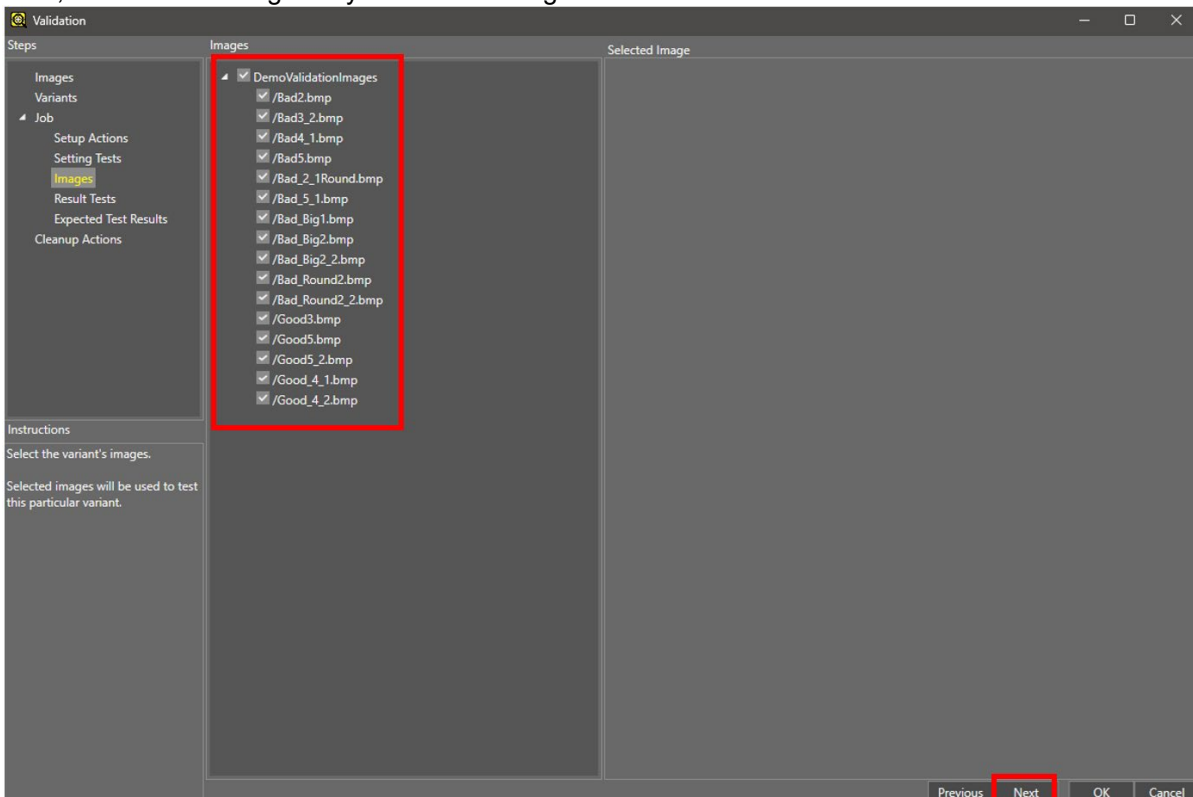
Next, create a Variant. Variants can be considered like job recipes. For instance, if you inspect multiple parts within the same job but the tolerances could change depending on them, you can set up your validation settings to correspond to multiple parts. For now, we will just use one and leave the default name of “Variant.”



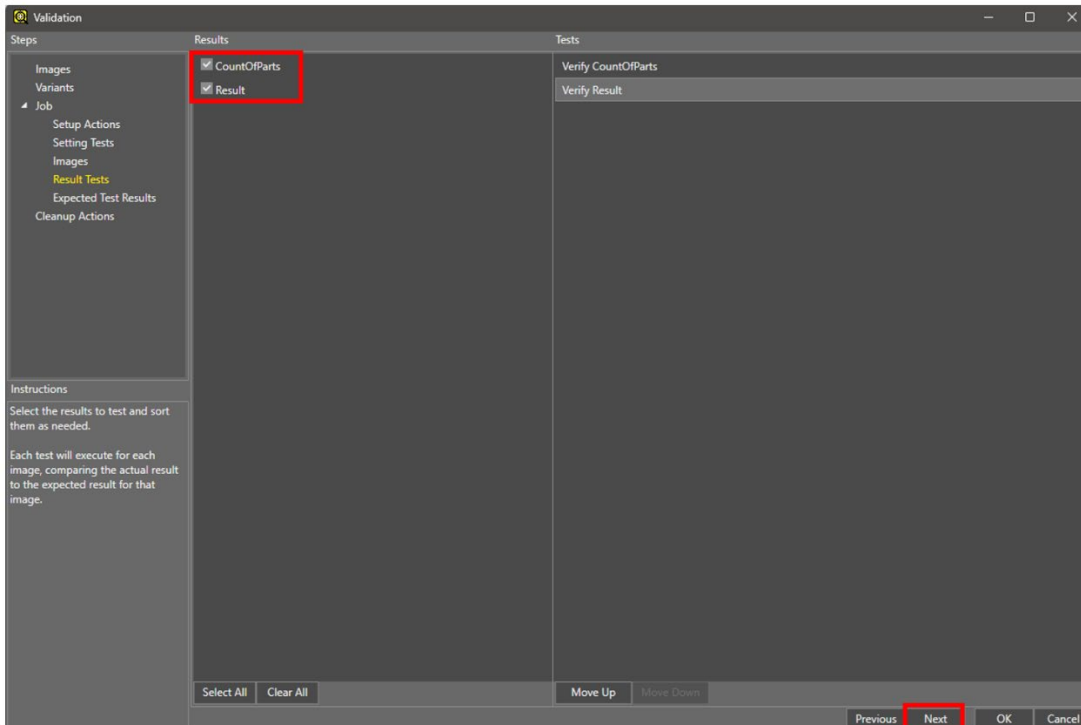
Once the variant is created, we will start with **Setup Actions**. Note that the variables we gave names to are placed in the list as checkboxes. From here, as we check them, we can set them up to have a specific input value.



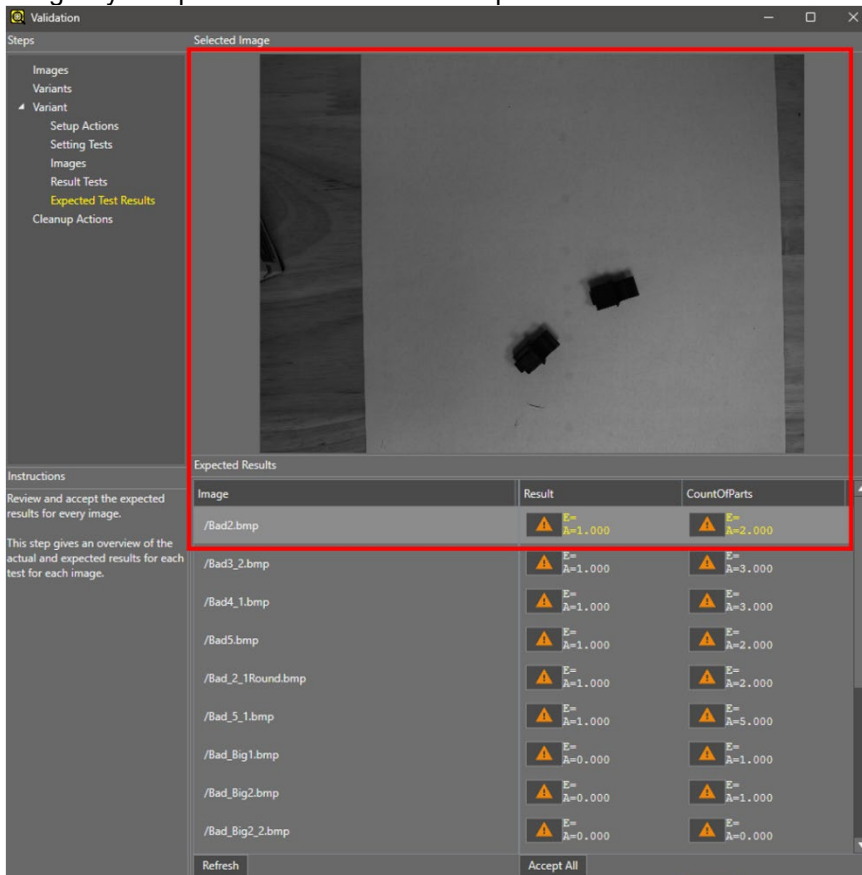
Next, select which images in your set to test against.



Next, we will set up the result tests. This automatically pulls in the non-editable variables that we assigned with names earlier.



Next, you will see the initial results coming from the tests for each of the images. Selecting the image will give you a preview of it towards the top.



You can accept the result from each variable either manually by clicking the individual alert icons or by clicking **Accept All**. This sets the expected values for the validation tests.

Expected Results

Image	Result	CountOfParts
/Bad2.bmp	✓ E=1.000 A=1.000	⚠ E= A=2.000
/Bad3_2.bmp	⚠ E= A=1.000	⚠ E= A=3.000
/Bad4_1.bmp	Click to accept the actual cell value as the new expected cell value	
/Bad5.bmp	⚠ E= A=1.000	⚠ E= A=2.000
/Bad_2_1Round.bmp	⚠ E= A=1.000	⚠ E= A=2.000
/Bad_5_1.bmp	⚠ E= A=1.000	⚠ E= A=5.000
/Bad_Big1.bmp	⚠ E= A=0.000	⚠ E= A=1.000
/Bad_Big2.bmp	⚠ E= A=0.000	⚠ E= A=1.000
/Bad_Big2_2.bmp	⚠ E= A=0.000	⚠ E= A=0.000

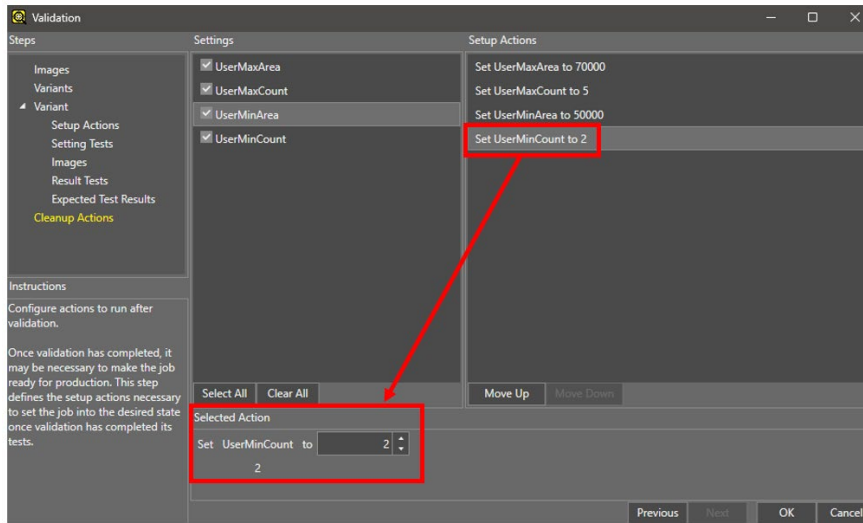
Refresh Accept All Previous Next OK Cancel

Expected Results

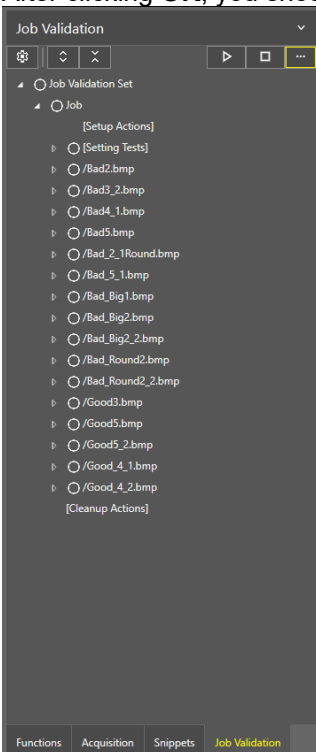
Image	Result	CountOfParts
/Bad2.bmp	✓ E=1.000 A=1.000	✓ E=2.000 A=2.000
/Bad3_2.bmp	✓ E=1.000 A=1.000	✓ E=3.000 A=3.000
/Bad4_1.bmp	✓ E=1.000 A=1.000	✓ E=3.000 A=3.000
/Bad5.bmp	✓ E=1.000 A=1.000	✓ E=2.000 A=2.000
/Bad_2_1Round.bmp	✓ E=1.000 A=1.000	✓ E=2.000 A=2.000
/Bad_5_1.bmp	✓ E=1.000 A=1.000	✓ E=5.000 A=5.000
/Bad_Big1.bmp	✓ E=0.000 A=0.000	✓ E=1.000 A=1.000
/Bad_Big2.bmp	✓ E=0.000 A=0.000	✓ E=1.000 A=1.000
/Bad_Big2_2.bmp	✓ E=0.000 A=0.000	✓ E=0.000 A=0.000

Refresh Accept All Previous Next OK Cancel

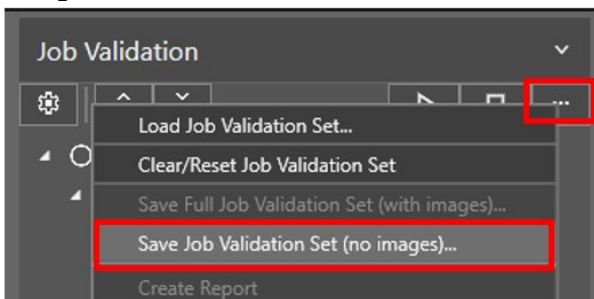
Lastly, if you changed any of the variables to have an initial value from the Setup Actions section, you can revert them back to the old value or some other value by using the Cleanup Actions section.



After clicking **OK**, you should have the validation setup within your side menu.

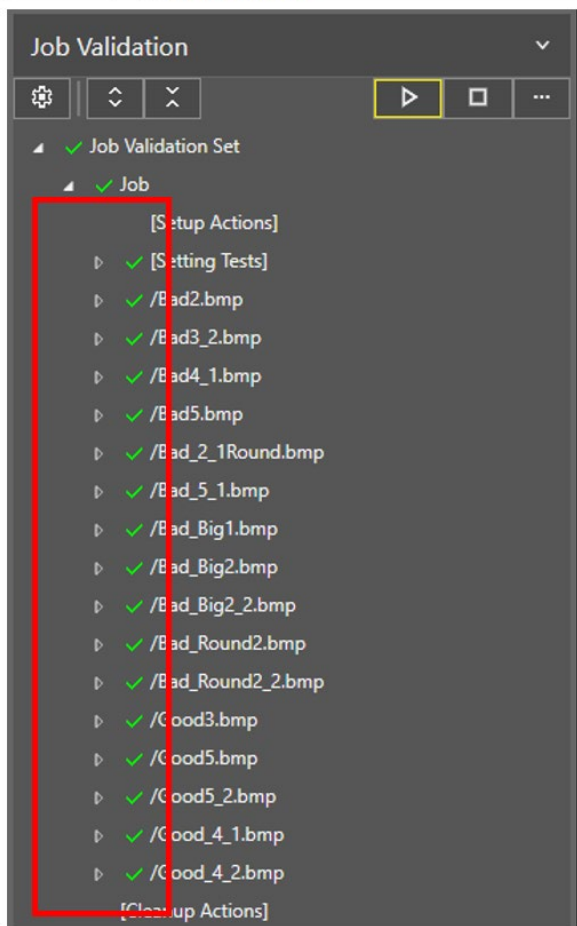
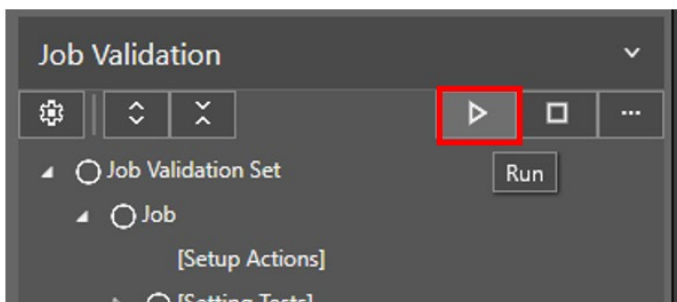


It's a good idea to save this now to the camera.

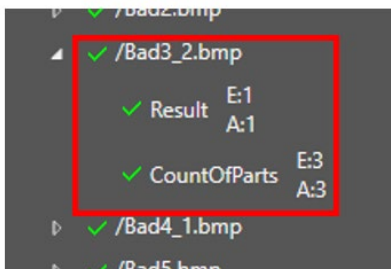


1.5 Run Job Validation

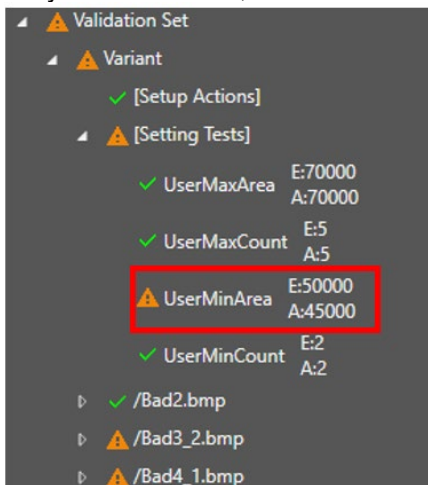
You can run the job validation by going to the Validation tab and clicking **Play**.



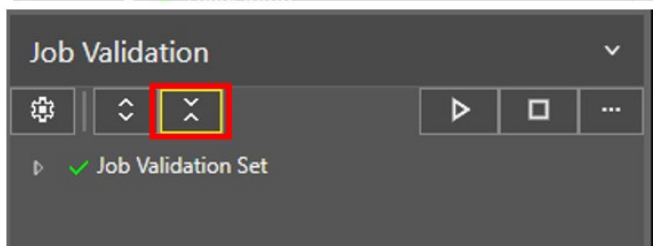
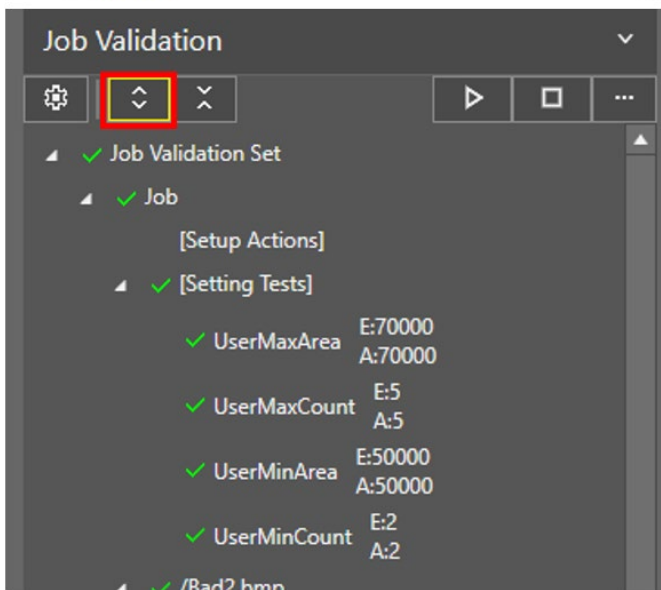
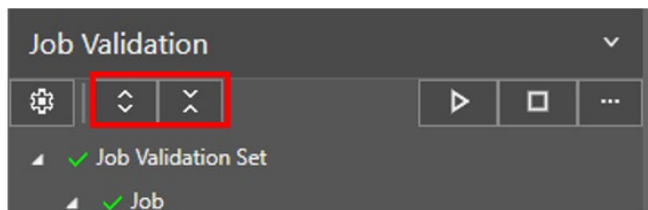
Every test will have a good/bad indicator on each image tested, giving you the individual results ("E" stands for expected value and "A" stands for actual value).



If any of the tests fail, the result will be shown with the Alert icon.

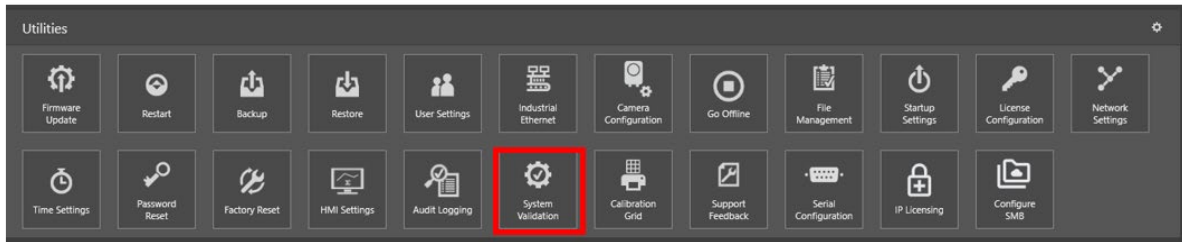


If you want to quickly expand or collapse the individual results, you can press the expand or collapse buttons at the top of the Job Validation panel.

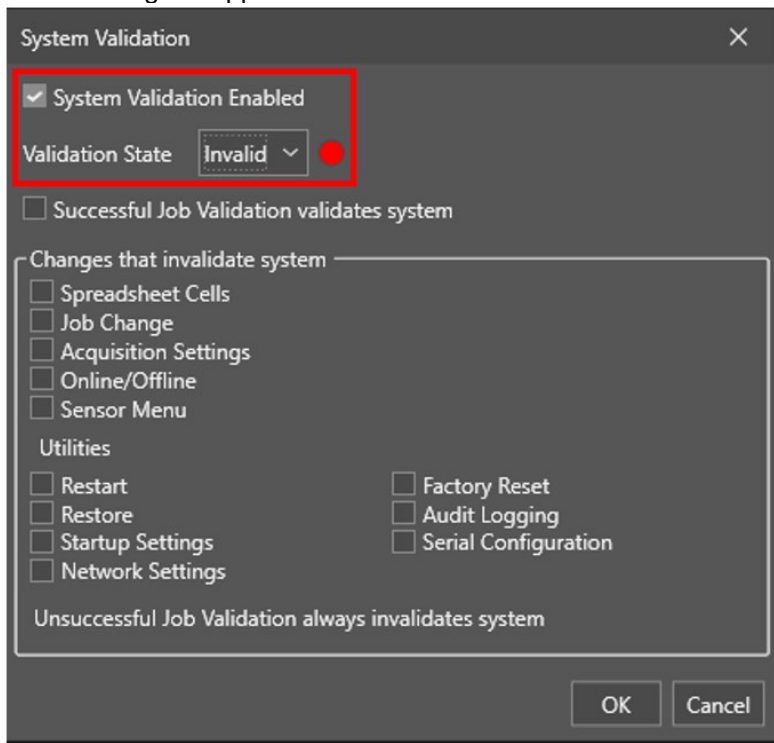


1.6 System Validation

System Validation is a new feature that can act as a flag to know if the system is still valid or if changes have been made that invalidate the system. Inside the utilities section of ISVS, you will see a System Validation option.



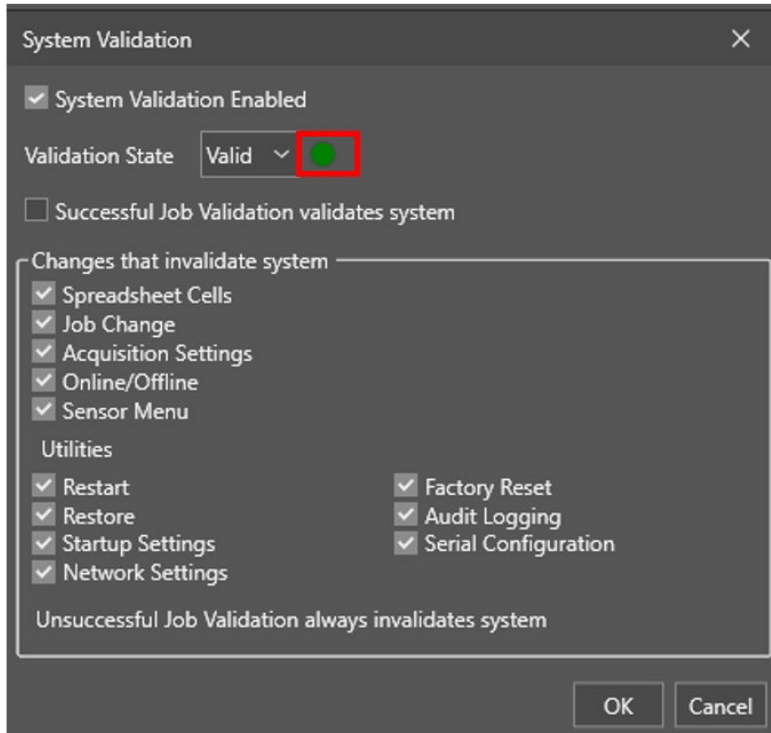
The following will appear.



From here, you can choose to enable/disable the validation option, set the current state, set up a successful job validation to validate the system, and lastly, set up what changes to the system/job would cause the system to become invalidated. Note, a Job Validation that is run and becomes unsuccessful will invalidate the system and this cannot be changed.

There are multiple ways to access the current state of the system.

First is checking the state via the System Validation within the ISVS Utilities.



The next is by using the web api to get the JSON formatted state.

You can view/test this by opening a web browser and entering in the following:

{IP address of camera}/api/system-validation/state

Pretty-print ☐

```
{"state":1}
```

Pretty-print ☐

```
{"state":0}
```

Pretty-print ☐

```
{"state":-1}
```

The states consist of the following

Value	State
1	Validated
0	Invalidated
-1	Disabled

To see what is tied to the validation state criteria, use the following API.

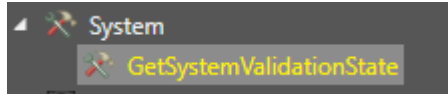
{IP address of camera}/api/system-validation/config

```

Pretty-print ☐

{"config":[{"event":"job-validation-failed","type":"job-validation"},{"event":"cell-expression","type":"cell-change"},
{"event":"cell-enabled-state","type":"cell-change"},{"event":"cell-name","type":"cell-change"},{"event":"cell-
comment","type":"cell-change"},{"event":"cell-protect-success","type":"cell-change"},{"event":"cell-protect-
failure","type":"cell-change"},{"event":"cell-unprotect-success","type":"cell-change"},{"event":"cell-unprotect-
failure","type":"cell-change"},{"event":"internal-data","type":"tool-change"},{"event":"training-data","type":"tool-
change"},{"event":"load-job","type":"job-change"},{"event":"save-job","type":"job-change"},{"event":"new-job","type":"job-
change"},{"event":"2D-acquisition","type":"acquisition-settings"},{"event":"3D-acquisition","type":"acquisition-settings"},
{"event":"online","type":"online-state-change"},{"event":"offline","type":"online-state-change"},
{"event":"dio","type":"sensor-menu"},{"event":"result-queue","type":"sensor-menu"},{"event":"custom-view","type":"sensor-
menu"},{"event":"job-status","type":"sensor-menu"},{"event":"config-syslog-forwarding","type":"audit-log"},
{"event":"shutdown","type":"audit-log"},{"event":"config-hostname","type":"network"},{"event":"config-interface-
parameter","type":"network"},{"event":"replace","type":"backup-restore"},{"event":"replicate","type":"backup-restore"},
{"event":"restore-backup-failure","type":"backup-restore"},{"event":"startup","type":"audit-log"},{"event":"startup-
settings","type":"utility"},{"event":"serial-configuration","type":"utility"},{"event":"initiated","type":"factory-reset"}]}
    
```

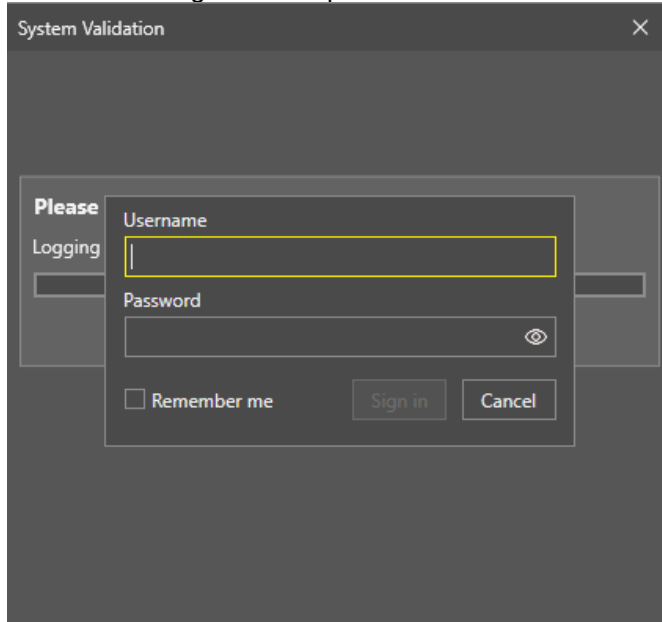
You can also access the Validation State within Spreadsheet.



GetSystemValidationState(\$A\$0)

1.7 User Control

By default, only a user with full privileges (such as admin) can access the ability to set the system validation settings. This helps ensure that others don't tamper with your validation settings.



Once logged in with the proper credentials, you can set the validation settings.