

# **Automation Boot Camp**

N-able RMM Version 2.11



Last updated: October 29, 2021

## **Table of Contents**

Lab 1 – Self-healing maintenance	3
Lab 2 – Proactive maintenance	7
Lab 3 – Proactive Maintenance	13
Lab 4 – Common objects	18
Lab 5 – Uploading and maintaining a policy	23
Lab 6 – Variables	26
Lab 7 – Input / output variables	33
Lab 8 – Custom check	38
Lab 9 – Using the run script object	42

## Lab 1 – Self-healing maintenance

			INPLIT
_			·
👰 Prompt			
Title: *	Maintenance Engine		0
Prompt: *	Hi, Can we do a quick mainter	nance to cleanup	o temporary files ? This should only take a minute and should not cause any slowdowns 📝
Wait (secs): *	600		Ø
Buttons: *	Yes, No		•
10.00		_	·
lt/Else			
		Variable: *	Prompt.Click Result
		Type:	string
		Condition: *	equals 🗸
		Value: *	Yes 🛷
🔁 Then		^	
			Else
	· ·	_	
📕 Delete	Temporary Files	^	· ·
<ul> <li>Delete 1</li> </ul>	emp Files for All Users		Eog ^
<ul> <li>Delete 1</li> </ul>	Femporary Internet Files for All U	Jsers	Message: * User has not accepted the maintenance. No maintenance occured 🤣
Delete (	Tookies		
	τ		

Create a basic disk clean-up policy to free space on a user's disk.

### **Estimated time**

10 minutes

### **Required resources**

Automation manager application

### Self-guided summary (advanced version)

For this lab, you will create an automation policy called LAB 01 - <YOURNAME>.

The goal is to create a policy that prompts the user for permission, and based on their answer, either runs the **Disk Cleanup** policy or not.

### **Detailed steps**

- 1. Create a new policy in automation manager
  - a. Name it LAB 01 <YOURNAME>, and describe it as "Lab 01 Disk Cleanup".
- 2. Add the required objects:
  - a. In the left menu, search for the **Prompt** object and drag it to the middle of the **Policy Builder**:



- b. In the left menu, search for the **If/Else** object and drag it to place it below the **Prompt** object.
- c. In the left menu, search for the **Delete Temporary Files** object and drag it into the **Then** section of the **If/Else** object.
- d. In the left menu, search for the **Log** object and drag it into the **Else** section of the **If/Else** object.
- 3. Enter the appropriate information in the objects' fields, and link required objects:
  - a. In the **Prompt** object, fill the fields as indicated below:
    - i. Title: Maintenance Engine
    - ii. Prompt: Hi, can we do a quick maintenance check?
    - iii. Wait: 600 seconds
    - iv. Buttons: Yes/No
  - b. In the **If/Else** object:

i. Click on the link icon ()

1

beside the **Variable** field.

ii. In the **Object Link** pop-up, under **Objects** click **Prompt**, then under **Ouptput parameters**, click **Click Result**, and click **OK** to save.

III Object Link
Objects
Objects
Prompt
0
Output parameters
Click Result

- iii. Select Equals, from the Condition drop-down list.
- iv. The object will look like this:

Variable: *	Prompt.Click Result	Ø
Type:	string	
Condition: *	equals	<b>~</b>
Value: *	Yes	- Ø

- c. In the **Log** object, enter a message noting that the user refused the maintenance.
- d. Run the policy by clicking the play 🔛 button.
- e. Save the file to your local drive for later use.

## **Optional steps**

- 1. You can run the policy on your computer multiple times by using **Yes/No** as options. You can also try not answering at all. You may need to reduce the wait to 10 seconds to test the **No** answer.
- 2. You can try to upload the policy to your server and run it on your own computer.

Notes:		
Where could I use this?		

## Lab 2 – Proactive maintenance

🔋 P	olicy Builder		
		INPUT	
	🛃 Is Service Runnin	g (PrintSpool)	
	Display or Service Nar	me: * Print Spooler	
1	lf	,	^
Var	iable: *	Is Service Running (PrintSpool).Conditional	
Тур	ie:	string	
Cor	ndition: *	does not equal 🗸	
Val	ue: *	True 🖉	
6	Then	^	
	🗾 Is Service Runnin	ng (Time) ^	
	Display or Service Nar	me: * Windows Time	
	📕 lf	^	
	Variable: *	Is Service Running (Time).Conditional	
	Type:	string	
	Condition: *	does not equal	
	Value: *	True 🛷	
	🛃 Then	_ ^	

🗣 Start Service 🔨
Service Name: * Windows Time 🛷
✓ If Disabled set Startup Type
Startup Type: * Automatic
🙀 Start Service 🔨
Service Name: * Print Spooler
If Disabled set Startup Type
Startup Type: * Select value V
👼 Wait
Wait (sec.): * 30 🔗
Js Service Running (Print Spooler 2)
Display or Service Name: * Print Spooler

🚰 lf	^
Variable: *	Is Service Running (Print Spooler 2).Conditional 🔗
Type:	string
Condition: *	does not equal
Value: *	True 🔗
	Fail Policy A No input required
	OUTPUT

Create a policy that can restart a windows service and look for a dependency service to ensure it is running. This is useful for line-of-business applications that have services with pre-requisites that must be started in sequence.

#### **Estimated time**

10 minutes

### **Required resources**

Automation manager application

### Self-guided summary (advanced version)

For this lab, you will create an Automation Policy called LAB02 – <YOURNAME>.

The goal is to check if the **Print Spooler** service is running. If it is not running, we need to check if the dependency service, **Windows Time**, is running. If it is running, we want to restart it, and if it is not running, we want to start it.

Then we need to start the **Print Spooler** service and after a five second wait time, we need to check if the service is running. If it is not running, we need to fail the policy execution to trigger a failed status.

### **Detailed steps**

- 1. Create a new policy in automation manager
  - a. Name it LAB02 <YOURNAME> and describe it as "Lab 02 Restart Windows Service".
- 2. Add the required objects (search for and drag the objects as you did in Lab 1):
  - a. Add Is Service Running object below Input of the Policy Builder.
    - i. Click the title bar for the object and add "(Print Spool)" to the object name.
      - ii. In the **Display or Service Name** field, enter "Print Spooler".
  - b. Add the If object:
    - i. Click the link icon at the right of the Variable field.
    - ii. In the Object Link pop-up, under Objects click Is Service Running.
    - iii. Under Ouptput parameters, click Conditional, and click OK to save.
    - iv. In the Condition field, select does not equal from the drop-down list.
    - v. In the **Value** field, enter "True". Note that this is case sensitive.
  - c. Add the following objects to the **Then** section of the **If** object:
    - i. Add the **Is service running** object:
      - 1. Click in the title bar and add "(time)" to the object name.
      - 2. In the Display or Service Name field, enter "Windows Time".
    - ii. Add the **If** object:
      - 1. Click the link icon at the right of the Variable field.
      - 2. In the **Object Link** pop-up, under **Objects** click **Is Service Running (time)**.
      - 3. Under **Ouptput parameters**, click **Conditional**, and click **OK** to save.
      - 4. In the **Condition** field, select **does not equal** from the drop-down list.

- 5. In the Value field, enter "True". Note that this is case sensitive.
- 6. In the **Then** section of this new **If** object, add the **Start Service** object.
- 7. In the Service Name field, enter "Windows Time".
- 8. Select the **If disabled set Startup Type** check box.
- 9. Select Automatic in the Startup Type dropdown list.
- d. After the second **If/Then** object, but still within the first **Then** object (this may be confusing, refer to the screenshot above if required), add the following objects:
  - i. Add the **Start Service** object.
    - 1. In the Service Name field, enter "Print spooler".
  - ii. Add the **Wait** object.
    - 1. In the **Wait (sec)** field, enter "30".
  - iii. Add the Is service running object.
    - 1. Click in the title bar and add "(Print Spool 2)" to the object name.
    - 2. In the **Display or Service Name** field, enter "Print Spooler".
  - iv. Add another **If** object.
    - 1. Click the link icon at the right of the **Variable** field.
    - 2. In the **Object Link** pop-up, under **Objects** click **Is Service Running (print spool 2)**.
    - 3. Under **Ouptput parameters**, click **Conditional**, and click **OK** to save.
    - 4. In the **Condition** field, select **does not equal** from the drop-down list.
    - 5. In the **Value** field, enter "True". Note that this is case sensitive.
    - 6. In the **Then** section of this new **If** object, add the **Fail Policy** object.
- 3. You can now run the policy and test it. Since we look at the Print Spooler and Windows Time services, this should not do anything as they should be running.
- 4. Save the file to your local drive for later use.

## **Optional steps**

- 1. You can try to open the "services.msc" console and stop the Print Spooler service and run the policy again your computer. You will notice that the policy will run and restart the service.
- 2. You can try to upload it to your server and configure as self-healing on a Print Spooler Windows monitoring service (ask for help if you want to configure this during the course).

Notes:		
Where could I use this?		

## Lab 3 – Proactive Maintenance

🐁 Policy Builder	
NPUT	
▼	
🔤 Folder Exists 🔷	
Folder: * Input Parameters. Folder where the files are to be cleaned up 🤡	
Sk WEIse	
Variable.* Folder Exists. Conditional	
Type: string	
Condition: * equals · · · · · · · · · · · · · · · · · · ·	
≨i Then ▲	
▼	
👫 Folder File Count 🔨	
Folder: * Input Parameters.Folder where the files are to be cleaned up	
Filter: * Input Parameters. File mask for files to be deleted	
i Recurse	
م الالتفاقية الالتفاقية المراجع ا	
Variable: * Input Parameters.Number of Most Recent Files to Keep 🛷	
Type: number Condition * evelow than	
Value: * Folder File Count. File Count.	
file file ^	
▼	
i≆i Math ∧	Si Else
Number 0: * Folder File Count. File Count 🔗	
Number 1:* Input Parameters.Number of Most Recent Files to Keep @	📲 Log ^
	Message: * Folder Does Not Exist, No Action Taken 🛷
🗸 🗤 . 🎽 Find Top N Oldest Files 🔷	▼
Folder: * Input Parameters.Folder where the files are to be cleaned up 🔗	Fail Policy
Filter: * Input Parameters.file mask for files to be deteted 🛷	No input required
Number of Files: * Mot Enough Files to Delete, Fail the Execution to notify the end-user 🖉	· ·
🕼 Fail Policy 🧄	
No input required List.* Find Top N Oldess Files. FileList	
🖉 Each 🥎	
v	
Delete File	
File: * ForEact.Find Top N Oldest.Files.Fields.File Name 3	
▼	
OUTPUT	

Create a policy that will look through a folder and keep a certain number of most recent files, ensuring that not all files are deleted. This is useful where backups stop running and you accidentally delete all files based on date, without noticing that new files are not being created.

### **Estimated time**

10 minutes

### **Required resources**

Automation manager application

### Self-guided summary (advanced version)

For this lab, you will create an automation policy called LAB 03 - <YOURNAME>.

The goal is to have a policy that has three input parameters:

- A count of the number of files to keep in the folder.
- The path of the folder to check.
- The file mask (typically \*.\*).

The policy would need to check if the folder exists, and if it doesn't, a log object should be used to output a "no update needed" state. If the folder exists, it needs to get a file count, do a math calculation to get the number of files to delete (total files – files to keep), then feed that result to an object to find the count of the number of oldest files.

The policy would then need to loop through all the files found and delete them.

### **Detailed steps**

- 1. Create a new policy in automation manager:
  - a. Name it LAB 03 <YOURNAME> and describe it as "Lab 03 Delete Old Files".
- 2. Click **INPUT** in the Policy Builder.

a. Create three Input Parameters that match the information below:

 Input Parameters and Global Variables				
Input Parameters	Global Variables			
Name	Display	Туре	Value	
filecounttokeep	Number of Most Recent Files to Keep	number	10	
folderpath	Folder where the files are to be cleaned up	string	c:\temp\testtokeep	
filemask	file mask for files to be deleted	string	*.*	

3. Add the required objects (search for and drag the objects as you did in the previous labs).

- a. Add the Folder Exists object.
  - i. Click the link icon at the right of the **Folder** field.
  - ii. In the Object Link pop-up, under Objects click Input Parameters.
  - iii. Under **Ouptput parameters**, click **Folder where the files are to be cleaned up**, and click **OK** to save.
- b. Add an If/Else object.
  - i. Click the link icon at the right of the **Variable** field.
  - ii. In the **Object Link** pop-up, under **Objects** click **Folder Exists**.
  - iii. Under Ouptput parameters, click Conditional, and click OK to save.
  - iv. In the **Condition** field, select **equals**.
  - v. In the Value field, enter True.
  - vi. The completed object will look like this:
- c. In the **Else** section, add a **Log** object.
  - i. In the Message field, enter "Folder does not exist, no action taken."
- d. In the Else section immediately after the Log object, add a Fail Policy object.
- e. In the **Then** section, add a **Folder File Count** object.
  - i. Click the link icon at the right of the Variable field.
    - ii. In the **Object Link** pop-up, under **Objects** click **Input Parameters**.
  - iii. Under **Ouptput parameters**, click **Folder where the files are to be cleaned up**, and click **OK** to save.
  - iv. Click the link icon at the right of the Filter field.
  - v. In the **Object Link** pop-up, under **Objects** click **Input Parameters**.
  - vi. Under **Ouptput parameters**, click **file mask for files to be deleted**, and click **OK** to save.
- f. In the same Then section immediately below the Folder File Count object, add another



#### If/Else object.

- i. Click the link icon at the right of the Variable field.
- ii. In the **Object Link** pop-up, under **Objects** click **Input Parameters**.
- iii. Under **Ouptput parameters**, click **Number of Most Recent Files to Keep**, and click **OK** to save.
- iv. In the Condition field, select greater than.
- v. Click the link icon at the right of the Value field.
- vi. In the **Object Link** pop-up, under **Objects** click **Folder File Count**.
- vii. Under **Ouptput parameters**, click **File Count**, and click **OK** to save.

viii. The completed object will look like this:



- g. In the **Then** section of this second **If/Else** object, add a **Log** object.
  - i. In the Message field, enter "Not enough files in the folder to require deletion. Failing policy."
- h. In the same Then section immediately below the Log object, add a Fail Policy object.
- i. In the Else section of this second If/Else object, add a Math object.
  - i. Click the link icon at the right of the **Number 0** field.
  - ii. In the **Object Link** pop-up, under **Objects** click **Folder File Count**.
  - iii. Under Ouptput parameters, click File Count, and click OK to save.
  - iv. Click the link icon at the right of the Number 1 field.
  - v. In the **Object Link** pop-up, under **Objects** click **Input Parameters**.
  - vi. Under **Ouptput parameters**, click **Number of Most Recent Files to Keep**, and click **OK** to save.
  - vii. In the Operation field, select Subtract.
- j. In the same **Else** section immediately below the **Math** object, add a **Find Top N Oldest Files** object.
  - i. Click the link icon at the right of the Folder field.
  - ii. In the Object Link pop-up, under Objects click Input Parameters.
  - iii. Under **Ouptput parameters**, click **Folder where the files are to be cleaned up**, and click **OK** to save.
  - iv. Click the link icon at the right of the **Filter** field.
  - v. In the Object Link pop-up, under Objects click Input Parameters.
  - vi. Under **Ouptput parameters**, click **file mask for files to be deleted**, and click **OK** to save.
  - vii. Click the link icon at the right of the **Number of Files** field.
  - viii. In the **Object Link** pop-up, under **Objects** click **Math**.
  - ix. Under Ouptput parameters, click NumberResult, and click OK to save.
- k. In the same **Else** section immediately below the **Find Top N Oldest Files** object, add a **For Each** object.
  - i. Click the link icon at the right of the List field.
  - ii. In the Object Link pop-up, under Objects click Find Top N Oldest Files.
  - iii. Under Ouptput parameters, click FileList, and click OK to save.
- I. In the Each section of this For Each object, add a Delete File object.
  - i. Click the link icon at the right of the **File** field.
    - ii. In the **Object Link** pop-up, under **Objects** click **For Each**.
    - iii. Under Ouptput parameters, click FileName, and click OK to save.
- 4. Save the file to your local drive for later use.

### **Optional steps**

- 1. You can create a temp folder in C:\test and add 50 miscellaneous files, then run the policy on that folder and only keep 30. You will notice that the oldest files are removed.
- 2. You can try to upload it to your server and run it on your own computer to see its output. Ensure that the folder exists or try running the policy on a non-existent folder to see the result.

lotes:
Vhere could I use this?

## Lab 4 – Common objects

	▼	
	Dis User Logged On	^
	Domain: * *	et l
	User Name: * *	Ø
/Else		
	Variable: * Is User Logged On.Cond	itional 🛷
	Type: string	
	Condition: * equals	<b>~</b>
	Value: * True	<i>I</i>
Then		^
📕 Folder Exist	s ^	
Folder: * C:\P	rogram Files (x86)\Adobe\Acrobat Reader DC\Reader 🥏	
		🗭 Else
🚰 if		^
Variable: *	Folder Exists.Conditional	
Type:	string	Log
Condition: *	equals	V Message: * Nobody's home
Value: *	True	
Then		▲ Is Application Installed
		Application: * Adobe Acrobat Reader DC
	<b>V</b>	· · · · · · · · · · · · · · · · · · ·
🗾 File Exists	^	<b>—</b>
File: * C:\Program I	Files (x86)\Adobe\Acrobat Reader DC\Reader\AcroRd32.exe 🔗	in territoria de la constante
	•	Variable: * Is Application Installed.Conditional
🛱 lf		Type: string
Variable: *	File Evists Conditional	Condition: * equais
Type:	string	Value: ^ Faise
Condition: *	equals 🗸	Filen ^
Value: *	True	• • • • • • • • • • • • • • • • • • •
📕 Then		🖕 Fail Policy 🔷
		No input required
📠 File Creation Dat	e ^	
File: * C:\Program F	iles (x86)\Adobe\Acrobat Reader DC\Reader\AcroRd32.exe 🛛 🔗	
	~	· · · · · · · · · · · · · · · · · · ·
	~	

Page | 18

Create a policy that highlights the usefulness of some of the more common objects, and show the output values and how to use them.

### **Estimated time**

10 minutes

### **Required resources**

Automation manager application

## Self-guided summary (advanced version)

For this lab, you will create an automation policy called LAB 04 - <YOURNAME>.

The end goal is to have a policy that follows this sequence:

- 1. Check if a user is logged on
  - a. If a user is logged on, we need to check if the following folder exists: C:\program files (x86)\adobe\acrobat reader dc\reader.
    - i. If the folder exists, check if the following file exists: *C*:\*program files* (*x86*)\*adobe*\*acrobat reader d*\*reader*\*acrord32.exe*.
      - 1. If the file exists, get the file creation date.
    - ii. If the folder doesn't exist, use a log to say "Nobody's home.", and then check if an application called *Adobe Acrobat Reader DC* is installed. If it is not installed, fail the policy.

### **Detailed steps**

- 1. Create a new policy in automation manager
  - a. Name it LAB 04 <YOURNAME> and describe it as "Lab 04 Common Objects".
- 2. Add the required objects (search for and drag the objects as you did in the previous labs).
  - a. Add the Is User Logged On object.
    - i. Enter an asterisk (\*) in both the **Domain** and User **Name fields**.
  - b. Add an If/Else object.
    - i. Click the link icon at the right of the Variable field.
    - ii. In the **Object Link** pop-up, under **Objects** click **Is User Logged On**.
    - iii. Under Ouptput parameters, click Conditional, and click OK to save.
    - iv. In the Condition field, select equals.
    - v. In the Value field, type True.

vi. The completed object will look like this:



- c. In the Then section, add a Folder Exists object.
  - i. In the **Folder** field, enter "C:\Program Files (x86)\Adobe\Acrobat Reader DC\Reader".
  - ii. In the same **Then** section immediately below the **Folder Exists** object, add an **If** object.
    - 1. Click the link icon at the right of the Variable field.
    - 2. In the Object Link pop-up, under Objects click Folder Exists.
    - 3. Under **Ouptput parameters**, click **Conditional**, and click **OK** to save.
    - 4. In the **Condition** field, select **equals**.
    - 5. In the **Value** field, type **True**.
    - 6. The completed object will look like this:

👼 lf		`
Variable: *	Folder Exists.Conditional	Ð
Туре:	string	
Condition: *	equals	~
Value: *	True	4

- iii. In the Then section of this same If object, add a File Exists object.
  - In the File field, enter "C:\Program Files (x86)\Adobe\Acrobat Reader DC\Reader\AcroRd32.exe".
- iv. In the same **Then** section immediately below the **File Exists** object, add another **If** object.
  - 1. Click the link icon at the right of the **Variable** field.
  - 2. In the Object Link pop-up, under Objects click File Exists.
  - 3. Under Ouptput parameters, click Conditional, and click OK to save.
  - 4. In the **Condition** field, select **equals**.
  - 5. In the Value field, type True.
  - 6. The completed object will look like this:

👼 If		^
Variable: *	File Exists.Conditional	\$
Туре:	string	
Condition: *	equals	× .
Value: *	True	\$

- v. In the Then section of this If object, add a File Creation Date object, and in the File field, enter: "C:\Program Files (x86)\Adobe\Acrobat Reader DC\Reader\AcroRd32.exe".
- d. Go back to the first **If** object and notice that the **Else** section is currently empty.
  - i. Add a Log object and in the Message field, enter "Nobody's home".
  - ii. Add an **Is Application Installed** object and in the **Application** field, enter "Adobe Acrobat Reader DC".
  - iii. Add an If object.
    - 1. Click the link icon at the right of the **Variable** field.
    - 2. In the **Object Link** pop-up, under **Objects** click **Is Application Installed.**
    - 3. Under **Ouptput parameters**, click **Conditional**, and click **OK** to save.
    - 4. In the **Condition** field, select **equals**.
    - 5. In the Value field, select False.
  - iv. In the Then section, add a Fail Policy object.
- 3. Save the file to your local drive for later use.

## **Optional steps**

- 1. This simple policy shows you some of the common objects. You can try to modify it to check for various folders and files on your computer and observe if the policy finds those folders and files.
- 2. You can try to upload it to your server and run it on your own computer.

Notes:	
Where could I use this?	

## Lab 5 – Uploading and maintaining a policy

### **Objective**

This lab goes in RMM and shows how to interact with automation policies. This is useful to understand how to upload/download amps, and how to run them.

### **Estimated time**

10 minutes

### **Required resources**

The automation manager application The Lab 01 automation policy file

### Self-guided summary (advanced version)

For this lab, you need to start with the LAB01 file.

Here's what you will need to do:

- 1. Upload the policy to the RMM dashboard.
- 2. View it in RMM under the automated tasks repository.
- 3. Finally, from RMM, download the AMP file.

#### **Steps**

- 1. Open the RMM dashboard in your browser.
- 2. Go to Settings > Script Manager.

New Edit Delete Download Script Wiew Code     Automation Manager OS Name Description A Usage Notes Script Ch Automate Windows Workstation CPU Load %     Yes No     Yes Windows CPU Info     No Yes Windows	script manager						
OS         Name         Description *         Usage Notes         Script Ch         Automate           Windows         Workstation CPU Load %         Yes         No         Yes           Windows         CPU Info         No         Yes           Windows         Reminder         No         Yes	🕂 New 💉 Edit	🛅 Delete 🛃 Download S	cript 📋 View Code		🛃 Auto	omation Manag	jer
Windows         Workstation CPU Load %         Yes         No         Yes           Windows         CPU Info         No         Yes           Windows         Reminder         No         Yes	OS	Name	Description *	Usage Notes Script Ch Automate.			
Windows         CPU Info         No         Yes           Windows         Reminder         No         Yes	Windows	Workstation CPU Load	1 %		Yes	No	*
Windows Reminder No Yes	Windows	CPU Info			No	Yes	
	Windows	Reminder			No	Yes	

- 3. Click **New** in the upper-left corner of the Script Manager.
- 4. Enter the information for the task:
  - a. Name (required)
  - b. **Description** What the task does.
  - c. Usage Notes How and when to use the task, what parameters need to be passed to it.

d. Default Timeout

Script Manager

senpernunger								9	900
🕂 New 💉 Edit 🍸	Script Manager						×	on Manage	er
OS	🕂 New 💉 Edit	🗊 Delete 🛃 Download	Script 📋 View Code		🛃 Automa	tion Manage	er	Autom	
Windows	OS	Name 🛎	Description	Usage Notes	Script	Autom		Yes	
Windows	Windows	Lab01 - Self-Healing 🧹			No	Yes	*	Yes	
Windows	Windows	Languard Scan			No	Yes		Vec	
Windows	Windows	List Printers			Yes	Yes		103	

- e. **Type** Select the following:
  - i. Script Check Used for monitoring checks.
  - ii. Automated Task Used for maintenance and remediation type tasks.
- f. **OS** The operating system for which the task is designed.
- g. File Upload Browse to the AMP file that will be uploaded to your local device.

Add User Defined Scri	ipts	
Name:	Lab 01	
Description:	This is the description for Lab 01	
Usage Notes:	This is the usage notes for Lab 01. Use this area to describe parameters required, etc.	
Default Timeout (seconds):	3600	
Туре:	Script Check	
	V Automated Task	
OS:	Vindows	
	Мас	
	Linux	
Upload a script		
File upload:	Select script Brow	se
	Supported script types: sh, js, vbs, cmd, bat, pl, php, py, rb, ps1, amp	
Disclaimer: Please be aware	that we are not responsible for script contents and any harmful effects they may have on your syst	tems.
	Save Can	cel

- h. Click **Save** to start the upload.
- 5. In the Script Manager scroll down to the bottom to find your uploaded task.
- 6. Download files:
  - a. In RMM, from the Script Manager, find your LAB 01 file. You can click the **Name** column to sort alphabetically.
  - b. Select your LAB 01 file and click **Download** from the toolbar.

#### **Important Notes**

- When uploading an automation policy from outside of your organization, it is highly recommended to open it in automation manager first and review what it does, for security reasons.
- Also, when opening an automation policy in the you may get an error about the version. This is normal if the policy was created on a more recent version of automation manager.

Page | 24

Notes:		
Where could I use this?		

## Lab 6 – Variables

## Lab 6.1 – Numbers

		🗉 Input Para	meters and Gl	obal Variabl	es
File Search ^		Input Parameter	s Global Varial	oles	
Folder: * c:\temp		Use Clabel Veria			
Filter: * *.txt		Use Global Varia	bie Assignment of		s to diobal variables.
Recurse		Name	Display	Туре	Value
•		fliecount	filecount	number	0
orEach ^					
* File Search.Files					
Each ^					
					ADD REMOVE
Math					
Number 0: * Global Variables.filecount					OK CAN
Number 1: * 1					
Operation: * Add		Output Para	motors		
~		Output Fara	Ineters		
Global Variable Assignment	Na	ame	Display	Туре	Value
				.)62	
Variable: * Global Variables.filecount	file	efound	file count found	number	Global Variables.file
Variable: * Global Variables.filecount	file	efound	file count found	number	Global Variables.file
Variable: *     Global Variables.filecount       Type:     number       Value: *     Math.NumberResult	file	efound	file count found	number	Global Variables.file
Variable: *     Global Variables.filecount       Type:     number       Value: *     Math.NumberResult       Type:     number	file	efound	file count found	number	Global Variables.file
Variable: * Global Variables:filecount  Variable: * Global Variables:filecount Type: number Value: * Math.NumberResult Value: * Math.NumberResult Value: * Value: *	file	efound	file count found	number	Global Variables.file
Variable: *     Global Variables.filecount       Type:     number       Value: *     Math.NumberResult       Type:     number	file	efound	file count found	number	Global Variables.file
Variable: *     Global Variables:filecount       Type:     number       Value: *     Math.NumberResult       Type:     number	file	efound	file count found	number	Global Variables.file
Variable: *     Global Variables.filecount       Type:     number       Value: *     Math.NumberResult       Type:     number	file	efound	file count found	number	Global Variables.file
Variable: * Global Variables.filecount 🖉 Type: number Value: * Math.NumberResult 🐼 Type: number	file	efound	file count found	number	Global Variables.file
Variable: * Global Variables.filecount 🖉 Type: number Value: * Math.NumberResult 🖉 Type: number	file	efound	file count found	number	Global Variables.file
Variable: * Global Variables.filecount <	file	efound	file count found	number	Global Variables.file

To review how to interact with numbers and use math objects to do calculations.

### **Estimated time**

10 minutes

### **Required resources**

Automation manager application

### Self-guided summary (advanced version)

For this lab, you will need to create an automation policy called LAB 06.1 – <YOURNAME>.

The goal is to have a policy that looks in a folder, counts the number of files based on search criteria (using global variables), and outputs the count to an output parameter.

#### **Steps**

- 1. Create a new policy in automation manager.
  - a. Name it Lab 06.1 numbers and describe it as "Lab 06.1".
- 2. Create a global variable
  - a. Click **INPUT** in the Policy Builder.
  - b. Click Global Variables.
  - c. Click ADD.
  - d. Enter the following:
    - i. In the **Name** field: "filecount".
    - ii. In the **Display** field: "filecount".
    - iii. In the **Type** dropdown list, select **number**.
    - iv. In the Value field: 0.
  - e. Click **OK** to save and **OK** again.
- 3. Add the required objects (search for and drag the objects as you did in the previous labs).
  - a. Add a File Search object.
    - i. In the **Folder** field, enter "C:\temp".
    - ii. In the Filter field, enter "\*.txt".
  - b. Add a For Each object.
    - i. Click the link icon at the right of the **List** field.
    - ii. In the **Object Link** pop-up, under **Objects** click **File Search**.
    - iii. Under Ouptput parameters, click Files, and click OK to save
  - c. In the **Each** section, add a **Math** object.
    - i. Click the link icon at the right of the **Number 0** field.
    - ii. In the **Object Link** pop-up, under **Objects** click **Global Variables**.
    - iii. Under Ouptput parameters, click filecount, and click OK to save.
    - iv. and click **OK** to save.

- v. In the **Number 1** field, enter 1.
- vi. Select Add from the Operation drop-down list.
- d. In the same **Each** section immediately below the **Math** object, add a **Global Variable Assignment** object.
  - i. Click the link icon at the right of the Variable field.
  - ii. In the **Object Link** pop-up, under **Output Parameters** click **filecount**.
  - iii. Click the link icon at the right of the **Value** field.
  - iv. In the **Object Link** pop-up, under **Objects** click **Math**.
  - v. Under **Ouptput parameters**, click **NumberResult**, and click **OK** to save.
- 4. Create an output parameter.
  - a. Click **OUTPUT** in the Policy Builder.
  - b. Click ADD.
  - c. Enter the following:
    - i. In the Name field: "ofilecount".
    - ii. In the **Display** field: "Total Files Found In Folder".
    - iii. Select Number from the Type drop-down list.
    - iv. Link the Value field to the filecount Global Variable.
  - d. Click **OK** to save and **OK** again to close.
- 5. Save the policy to your computer.

## Lab 6.2 – Dates

Policy Builder	
	INPUT
🛃 Get Date	
Custom Date Forma	it: MMMM, yyyy
👼 File Crea	ation Date
File: * C:\	Temp\test1.txt
崖 Date Arithm	netic ^
Date: *	6/14/2019 1:38:56 PM 🗘 📅
	Link to a parameter
<ul> <li>Now</li> </ul>	
Operation: *	Subtract 🗸
Time Units: *	Days 🗸
Time Span: *	1 🖉
🚝 Date Differe	nce ^
First Date: *	6/14/2019 1:40:33 PM 🗘 🛗
	Get Date.Date Result 🔗
Now	
Second Date: *	6/14/2019 1:40:33 PM 👙 🛗
	File Creation Date.Date
Now	

🚰 If		^
Variable: *	Date Arithmetic.Resulting Date	
Type:	date	
Condition: *	greater than	
Value: *	File Creation Date.Date	
🚰 Then	^	Ľ
		1
🛃 Log	^	
Message: *	test 🛷	
	Ŧ	
	T	

This is meant to review the usage of date variables and how to do calculations on them

### **Estimated time**

10 minutes

### **Required resources**

Automation manager application

A text file created in C:\temp\test1.txt, or another file that can be used to compare dates.

### Self-guided summary (advanced version)

For this lab, you will need to create an automation policy called LAB 06.2 - <YOURNAME>.

The goal is to create a policy that retrieves a file creation date (C:\temp\file1.txt), then compares it to see if it is more than one day old.

Note that there are two objects that will do that calculation. Try to use both.

### **Steps**

- 1. Create a new policy in automation manager
  - a. Name it LAB 06.2 <YOURNAME>. and describe it as "Lab 06.2 Date".
  - b. Add the required objects (search for and drag the objects as you did in the previous labs).
    - i. Add a Get Date object.
    - ii. Add a File Creation Date object.
      - 1. In the **File** field, enter "C:\temp\test1.txt" or any file you put here to get a valid date.
    - iii. Add a Date Arithmetic object.
      - 1. Select the **Now** check box.
      - 2. Select **Subtract** from the **Operation** drop-down list.
      - 3. Select **Days** from the **Time Units** drop-down list.
      - 4. In the **Time Span** field, enter "1".
    - iv. Add a Date Difference object.
      - 1. For the **First Date**, link to the **Get Date** object and the **Date Result** Output Parameter.
      - 2. For the **Second Date**, link to the **File Creation Date** object and the **Date** Output Parameter.
      - 3. Select **Days** from the **Time Units** drop-down list.
      - v. Add an **If** object.
        - 1. Link the **Variable** to the **Date Arithmetic** object and the **Resulting Date** Output Parameter.
        - 2. Select greater than from the Condition drop-down list.
        - 3. Link the Value to the File Creation Date Object and the Date Output

Parameter.

- vi. In the **Then** section, add a **Log** object and enter "test" in the **Message** field.
- 2. Save the policy to your computer.

Notes:
Where could I use this?

## Lab 7 – Input / output variables

	INPUT		input raia	meters and Git		0103
🕞 File S	earch		Input Parameter	s Global Variabl	es	
Folder: *	c:\temp	1	Name	Display	Туре	Value
Filter: *	*.txt	Ð	ifilepath	File Path To Look	string	c:\temp
Recu	rse	_				
rEach		^				
	File Search.Files	ŵ				
Each		^				ADD REMO
	~					
if 🖁		^				
		(I)				OK (
Variable: *	Global Variables.filecount	~				
Variable: * Type:	Global Variables.filecount					
Variable: * Type: Condition: *	Global Variables.filecount number less than or equal to	×	III Input Par	ameters and G	obal Varia	bles
Variable: * Type: Condition: * Value: *	Global Variables.filecount number less than or equal to 10	* \$	💠 Input Par	ameters and Gl	obal Varia	bles
'ariable: * iype: iondition: * 'alue: * <b>ialue: *</b>	Global Variables.filecount number less than or equal to 10	♥ ₽	Input Para	ameters and Gl	obal Varia	bles
ariable: * ype: ondition: * alue: *	Global Variables.filecount number less than or equal to 10	<ul> <li>✓</li> <li>Ø</li> <li>∧</li> </ul>	Input Para Input Paramete Use Global Var	ameters and Gl ers Global Variat riable Assignment of	obal Varia Nes	bles lues to Global Variabl
ariable: * ope: ondition: * Hue: * Then	Global Variables.filecount number less than or equal to 10 String	<ul> <li></li> <li></li> <li></li> <li></li> <li></li> </ul>	Input Para Input Paramete Use Global Van Name	ameters and Gl ers Global Variat riable Assignment of Display	obal Varia Jes Ject to set val	bles lues to Global Variabl Value
/ariable: * (ype: (alue: * Then Format Input 0: *	Global Variables.filesount number less than or equal to 10 String Global Variables.filesfound	<ul> <li></li> <li><td>Input Parameter Input Parameter Use Global Van Name filecount</td><td>ameters and Gl ers Global Variat riable Assignment of Display filecount</td><td>obal Varia oles ject to set val Type number</td><td>bles lues to Global Variabl Value 0</td></li></ul>	Input Parameter Input Parameter Use Global Van Name filecount	ameters and Gl ers Global Variat riable Assignment of Display filecount	obal Varia oles ject to set val Type number	bles lues to Global Variabl Value 0

OK CANCEL

33© 2021 N-able Solutions ULC and N-able Technologies Ltd. All rights reserved.

input i.	ForEach.File Search.Files.Path
Text: {	0), {1}
📕 Global V	/ariable Assignment ^
Variable: *	Global Variables.filesfound
Type:	string
Value: *	Format String.FormattedString
Type:	string
🏓 Math	^
Number 0: *	Global Variables.filecount 🔗
Number 0: * Number 1: *	Global Variables.filecount 🔗
Number 0: * Number 1: * Operation: *	Global Variables.filecount
Number 0: * Number 1: * Operation: *	Global Variables.filecount
Number 0: * Number 1: * Operation: *	Global Variables.filecount
Number 0: * Number 1: * Operation: *	Global Variables.filecount
Number 0: * Number 1: * Operation: *	Global Variables.filecount     Image: Constraint of the second seco
Number 0: * Number 1: * Operation: * Global V Variable: * Type:	Global Variables.filecount
Number 0: * Number 1: * Operation: * <b>Global V</b> Variable: * Type: Value: *	Global Variables.filecount

#### III Output Parameters

Х

Name	Display	Туре	Value
ofilecount	Total Files Found In	number	Global Variables.filecour
ofilesfound	First 10 files found	string	Global Variables.filesfou
			ADD REMOVE
			OK CANCEL

To better understand how to use input/output variables in automation manager and RMM.

### **Estimated time**

10 minutes

### **Required resources**

Automation manager application

### Self-guided summary (advanced version)

For this lab, you will need to start from the LAB 6.1 policy file.

The goal is to modify the LAB 6.1 policy to use a dynamic file path as input, and to use that in the folder object, and then leverage a second global variable to append the first ten file names found, to an output parameter.

#### **Steps**

- 1. Open the automation policy LAB 6.1.
- 2. Save it as LAB 7 <YOURNAME>.
- 3. Click **Options** then **Properties**. Change name and description to be LAB 7.
- 4. Add an Input Parameter.
  - a. Click **INPUT** in the Policy Builder.
  - b. Click ADD.
  - c. Click the Input Parameters tab.
  - d. Enter the following:
    - i. In the **Name** field: "filepath".
    - ii. In the **Display** field: "file path to look into".
    - iii. In the **Type** drop-down list, select string.
    - iv. In the **Value** field: "C:\temp".
  - e. Click OK.
- 5. Add a Global Variable.
  - a. Click **INPUT** in the Policy Builder.
  - b. Click the Global Variables tab.
  - c. Click ADD.
  - d. Enter the following:
    - i. In the Name field: "filesfound".
    - ii. In the **Display** field: "filesfound".
    - iii. In the Type drop-down list, select string.
    - iv. Leave the **Value** field empty.
  - e. Click **OK** to save, and **OK** again.
- 6. Add the following objects:

- a. In the **Each** section of the **For Each** object immediately before the **Math** object, add an **If** object.
  - i. In the **If** object, link the **Variable** field to the **Global Variables** object and the **filecount** Output Parameter.
  - ii. Select less than or equal to from the Condition drop-down list.
  - iii. In the **Value** field, enter 10.
    - This will go in the **Then** section only if it has found 10 or less files.
- b. In the **Then** section, add a **Format String** object.
  - i. Link the **Input 0** field to the **Global Variables** object and the **filesfound** Output Parameter.
  - ii. Link the Input 1 field to the foreach object and the path Output Parameter.
  - iii. In the **Text** field, enter {0}, {1}. This will format the string and concatenate the two strings together separated by a comma.
- c. Add a Global Variable Assignment object immediately below the Format String object.
  - i. Link the **Variable** field to the **Global Variables** object and the **filesfound** Output Parameter.
  - ii. Link the **Value** field to the **Format String** object and the **FormattedString** Output Parameter.
- 7. Add an Output Parameter.
  - a. Click OUTPUT in the Policy Builder.
  - b. Click **ADD**.
  - c. Enter the following:
    - i. In the Name field: "ofilesfound".
    - ii. In the **Display** field: "First 10 files found in the folder that matched the criteria".
    - iii. In the **Type** drop-down list, select **string**.
    - iv. Link the Value field to the Global Variables object and the filesfound Output Parameter.
    - v. Click **OK** to save and click **OK** twice more.
- 8. Run the policy locally.
- 9. Save the policy to your local drive.

# Lab 8 – Custom check



### **Objective**

This lab is intended to review the process to create a custom check in RMM.

### **Estimated time**

10 minutes

### **Required resources**

Automation manager application

## Self-guided summary (advanced version)

For this lab, you will need to start with the LAB 7 file.

The goal is to create a custom check from the LAB 7 file.

You will upload the policy to RMM, add the custom check to a device and run the check to observe the output.

### **Steps**

1. Upload and run the policy in RMM.

Edit User Defined Scri	ipts	
Name: Description:	Lab 07	
Usage Notes:		1
Default Timeout (seconds):	3600	Ĩ
Type:	Script Chark	
.,,=	Automated Task	
OS:	Vindows	
	Mac Mac	
	Linux	
Upload a script		
File upload:	Upload new script version Brows	se
	Supported script types: sh, js, vbs, cmd, bat, pl, php, py, rb, ps1, amp	
Disclaimer: Please be aware	e that we are not responsible for script contents and any harmful effects they may have on your syst	ems.
	Save Can	cel

- a. Log in to RMM and upload the Lab 07 policy according to the steps in Lab #5.
  - i. Select Script Check for the Type.

	I         Page         1         I									
Su	ummar	γ	Outages	C	Checks Notes Tasks Assets	Patches	Antivirus	Backup	Web	
+	Add Ch	neck <del>-</del>	Check	-						
ŀ	Add	24x7 (	Check 🕨	0	Description					More Information
	Add	DSC C	heck 🕨		Antivirus Update Check					Monitoring Status is No Sta
	~	ŀ	*	2	Backup & Recovery Selected Size Check					Total: 49.04GB, Free: 32.72
	~	ŀ	¢.,	2	Backup Check	ng Agent Web	Protection			Status RUNNING
	~	ŀ	$\diamond_{\circ}$		Critical Events Check	ntroller				Status RUNNING
	~	ŀ	¢ <sub>o</sub>	Ē	Drive Space Change Check	ess Launcher				Status RUNNING
	~	ŀ	¢.,		Event Log Check	Endpoint Inte	egration Service	9		Status RUNNING
	1	ŀ	¢ <sub>o</sub>	+	Failed Login Check	Endpoint Mas	ster Service			Status RUNNING
	~	ŀ	¢.,		Physical Disk Health Check	Endpoint Pro	tected Service			Status RUNNING
	~	ŀ	¢.,		Script Check	Endpoint Red	lline Service			Status RUNNING
	~	ŀ	$\diamond_{\circ}$		SNMP Check	Endpoint Sec	urity Service			Status RUNNING
	~	ŀ	٩, ,	NIPIP C	Windows Service Check - Managed Antiviru	s Endpoint Upo	date Service			Status RUNNING
	<ul> <li></li></ul>	$\phi_{\alpha}$	Wind	lows Se	ervice Check - Managed Antivirus Endpoint Update Se	vice		Status R	UNNING	

- c. Click Add Check > Add DSC Check > Script Check.
- d. Select the Lab 07 task from the list (search or find it in the User Defined section).
- e. Click Next.

Add Script Check	
Q Lab 07	×
C Lab 07  User Defined Lab 07 Lab 07 Lab 07 - Line Break Test	×
Please select a script to see additional details.	
	Next Cancel



Lab 07	(
lert Settings 🔻	Multiple Devices
Script Parameters	
file path to look for:	
c:\temp	
Script timeout (Range: 1 - 150 seconds):	
10	
Nederinez	
Jisclaimer Nease be aware that we are not responsible for user de	ined script contents and
<b>Disclaimer</b> Nease be aware that we are not responsible for user det any harmful effects they may have on your system. Sasign a Task after creating the Check	ined script contents and
Disclaimer Nease be aware that we are not responsible for user det any harmful effects they may have on your system. Sasign a Task after creating the Check Dick here for help setting up the Script Check	ined script contents and
Disclaimer Nease be aware that we are not responsible for user det any harmful effects they may have on your system. Assign a Task after creating the Check Dick here for help setting up the Script Check	ined script contents and
Disclaimer Nease be aware that we are not responsible for user det my harmful effects they may have on your system. Assign a Task after creating the Check Lick here for help setting up the Script Check	ined script contents and
Disclaimer Nease be aware that we are not responsible for user de my harmful effects they may have on your system. Assign a Task after creating the Check Lick here for help setting up the Script Check	ined script contents and

f. Enter the file path to look for parameter.

#### g. Click Finish.

h. Right-click the device in the upper pane and click **Run Checks.** 

Se	rvers	Workstations	Mixed Mobile D	evices	Network Devices	Services	Networks				
Wo	Workstation 👻 🗴 Take Control 🕎 Remote Background 🗸										
	Ë	Client	Site	Wor	Workstation			Operating System			
	Ĩx.	A1 Demo Cor	p Main Office	A1D	EMO-02-W7		Windows 7	Enterprise Editi	on, 64-bit Service Pack 1 (build 7601)		
		A1 Demo Corp	p Main Office		EMO 14/10		Windows 1	0 Enterprise N, (	64-bit (build 18362)		
	Ĩx.	A1 Demo Cor	p Main Office		Add Workstation Note		Windows 7	Enterprise Editi	on, 64-bit Service Pack 1 (build 7601)		
	i.	A1 Demo Cor	p Main Office	6	Pending Workstation Actio	ns b	Windows 1	0 Enterprise N,	64-bit (build 18362)		
	i.	A1 Demo Corp	p Main Office		Remote Background	•	Windows 1	0 Enterprise, 64	-bit (build 17763)		
ú		A1 Demo Corp	p Main Office	(h)	Debaat	h.	macOS 10.	14.5 build 18F13	32		
	Ī.	A1 Demo Corp	p Main Office	<u> </u>	Rebool		Windows 1	0 Enterprise, 64	-bit (build 17763)		
	E	A1 D C	Main 046	3	Backup & Recovery	•	148-d 4	0 F-t	CA Lik (L.:13 10040)		
14	4 Pag	e 1 of 1		۲	Managed Antivirus	Þ					
Su	mmary	Outages C	hecks Notes 1		Web Protection	▶ ntivirus	Backup	Web			
+	Add Check	Check			Patch Management	Þ					
	-	ii D	escription	0	Run Checks				More Information		
	÷ 🗄	) 📃 🛛 S	cript Check - Lab 07 - if	il	Task	•			Awaiting synchronization		

i. Wait 30 seconds and refresh the lower pane (Refresh icon in the upper-right corner of the lower pane).

same y	innapr.	Marin	Lans	inter [	Apres	Lonion.	sairies	indep:	100			2.12
4 ALCORE												action \$
	1	Incrusive.				Per la	- Alizability			Sec. Ber	Colympical Data	
1 FT	57	And in the second		dana i	interest in		-	-				

j. When the **More Information** value for the Lab 07 check reads **Awaiting first run**, continue to step k.

Sum	nmar	y (	Outag	jes	Checks Notes Tasks Assets Pate	hes Antivirus	Backup Web
- Ac	dd Ch	eck 🕶	Che	ck 🕶			
				Ë	Description	More Information	
	÷	1	>_		Script Check - Lab 07 - ifilepath:"c:\temp"	Awaiting first run	

- k. Right click the Script Check Lab 07 and click **Run Check**.
- I. When the **More Information** value for the Lab 07 check reads **More information**, continue to step m.

Summary	Outag	jes	Checks Notes Tasks Assets Pate	hes Antivirus	Backup Web
H Add Check	- Che	eck 🕶			
-		Ë	Description	More Information	
✓ É	) >		Script Check - Lab 07 - ifilepath:"c:\temp"	More information	

- m. Click on the **More information** link for Script Check Lab 07 to view the output.
- n. The output variables will be used in the next two labs.



## Lab 9 – Using the run script object

Policy Builder				
			INPU	T
Run Pow	verShell So	ript		
Input Para	meters	Script	Outp	ut Parameters
Name	Display	/ T <u>j</u>	ype	Value
scrandomm	ir scrando	mmir nu	umber	1
scrandomm	a: scrando	mma: nu	umber	60
				ADD EDIT REMOVE
📄 w	/ait			^
Wait	(sec.): *	Run Pow	erShell S	Script.scoutrandomval 🔗

To cover the basics of using your own script in a run script object with input/output variables.

### **Estimated time**

10 minutes

### **Required resources**

Automation manager application

### **Steps**

- 1. Create a new policy in automation manager
  - a. Name it Lab 10 <YOURNAME> and describe it as "Lab 10 Run Script".
- 2. Add two Input Parameter as shown in the screenshot below:

 Input Parameters and Global Variables ×				
Input Parameters	Global Variables			
Name	Display	Туре	Value	
inrandommin	Minimum Random Value	number	1	
inrandommax	Maximum Random Value	number	50	

- 3. Add the required objects (search for and drag the objects as you did in the previous labs).a. Add a **Run Powershell Script** object.
  - i. Create 2 Input Parameters as shown in the screenshot below:

III Edit Input Parameter "scrandommin"     ×       III Edit Input Parameter "scrandommax"     ×				
Name: *	scrandommax			
Display: *	scrandommax			
Type: *	number	$\sim$		
Value: *	Input Parameters.Maximum Random Value	Ø		

Page | 43

ii. Create one Output Parameter as shown in the screenshot below:

-

💠 Edit Output Parameter "scoutrandomval" 🛛 🗙				
Name: *	scoutrandomval			
Display: *	scoutrandomval			
Type: *	number	~		
	OK CANCE	:L		

- iii. Click the Script tab, click EDIT and Copy/paste the following script: \$randommin = \$scrandommin \$randommax = \$scrandommax \$irandomval = get-random - Minimum \$randommin - Maximum \$randommax \$scoutrandomval = \$irandomval
- 4. Add a **Wait** object and link the **Wait (sec)** field to the output variable of the script.

Notes:		
Where could I use this?		

N-able (formerly SolarWinds MSP) empowers managed services providers (MSPs) to help small and medium enterprises navigate the digital evolution. With a flexible technology platform and powerful integrations, we make it easy for MSPs to monitor, manage, and protect their end customer systems, data, and networks. Our growing portfolio of security, automation, and backup and recovery solutions is built for IT services management professionals. N-able simplifies complex ecosystems and enables customers to solve their most pressing challenges. We provide extensive, proactive support—through enriching partner programs, hands-on training, and growth resources—to help MSPs deliver exceptional value and achieve success at scale. n-able.com

The N-ABLE, N-CENTRAL, and other N-able trademarks and logos are the exclusive property of N-able Solutions ULC and N-able Technologies Ltd. and may be common law marks, are registered, or are pending registration with the U.S. Patent and Trademark Office and with other countries. All other trademarks mentioned herein are used for identification purposes only and are trademarks (and may be registered trademarks) of their respective companies.