Agent Procedure

Identifies USB External Storage

For Exclusion in Creating Alarms

Guide

August 4, 2020

Copyright Agreement

The purchase and use of all Software and Services is subject to the Agreement as defined in Kaseya's "Click-Accept" EULATOS as updated from time to time by Kaseya at http://www.kaseya.com/legal.aspx. If Customer does not agree with the Agreement, please do not install, use or purchase any Software and Services from Kaseya as continued use of the Software or Services indicates Customer's acceptance of the Agreement."

USB Storage Exclusion For Low Disk Alarms (Ver A)

The Ver A labeled solution defines a monitor set that will test Windows OS % free space for drive letters C: - I:. Import the monitor set and agent procedure included in this solution package.

Import all items from System – Import Center. The File to import is PS__USB_Storage_Alarm_Exclusion.xml.

In the Agent Procedure Module in the Shared – Import Center folder:

```
PS-External USB Drives Exclusion (Rev A)
PS-USB Flash Drive Exclusion (Rev B)
```

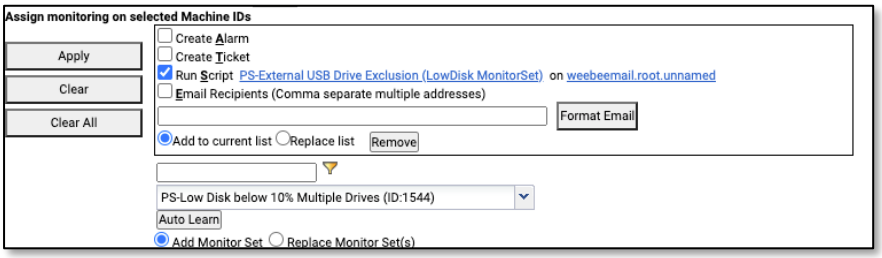
In the Monitor Module in the AutoExchange – PS-LowDisk folder:

```
PS-Low Disk less than 10% (Ver A. Multiple Drives)
PS-Low Disk less than 10% (Ver B. All Drives)
```

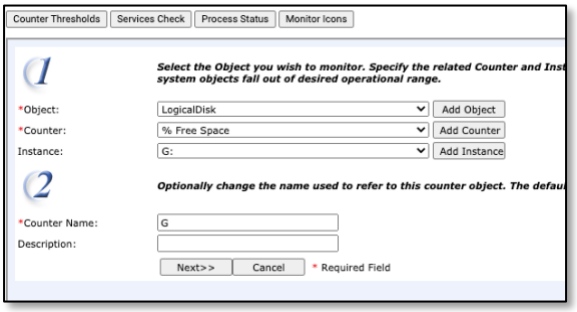
Ver A will exclude all USB storage devices. This includes USB Flash Drives and USB External Hard Drives.

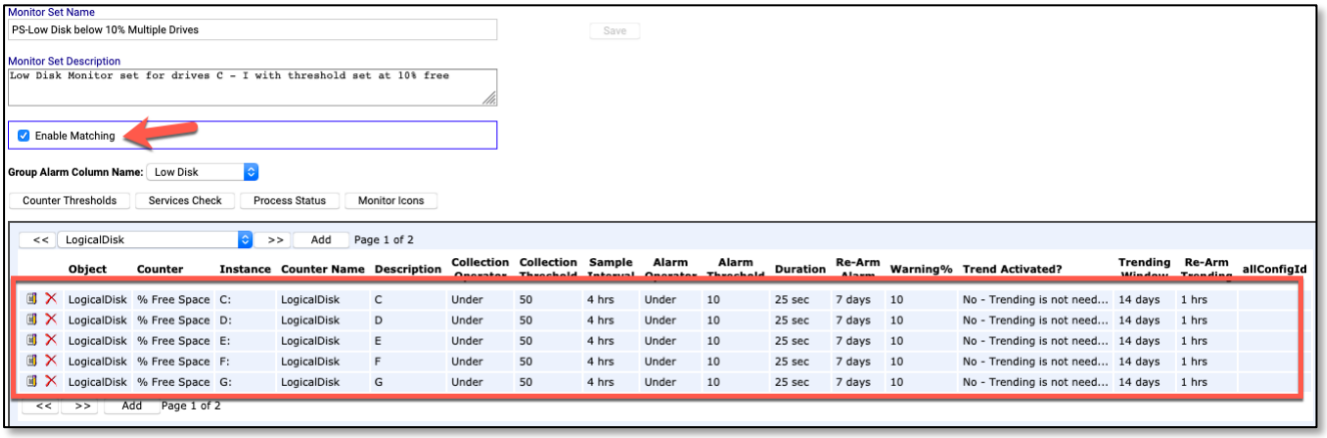
Monitor Set

Assign imported Monitor Set to Windows OS endpoint. Set alarm notification to Run Script and point to the imported agent procedure.



Monitor Set will check for drive letters C: through I: for % Free Space less than 10%. Modify the alarm threshold or add new drive letter instance if necessary.





Agent Procedure

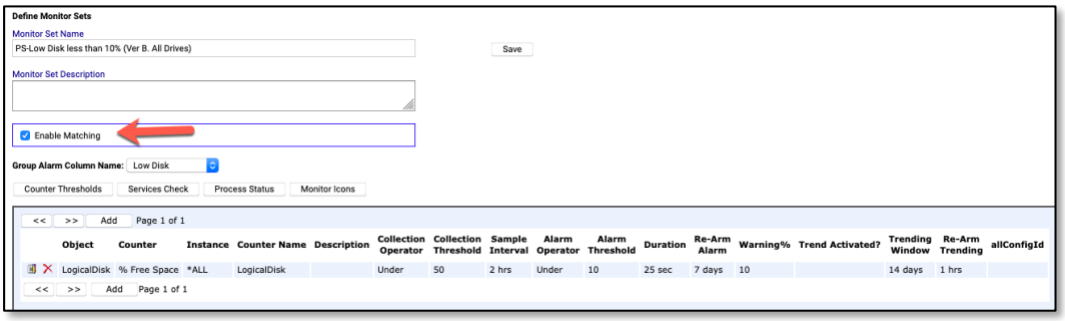
The agent procedure leverages variables from the Alarm. The alarm properties allows to check individual alarms, whether the drive letter is a USB storage then act accordingly.
The PowerShell script to test whether a drive is a USB storage device.

```
try {(Get-Partition -ErrorAction SilentlyContinue -DiskNumber (Get-Disk | Where-Object -FilterScript {$_.Bustype -Eq 'USB'}).Number).DriveLetter} catch {'NO USB DRIVE DETECTED'}
```

USB Storage Exclusion For Low Disk Alarms (Ver B)

Monitor Set

Ver B will monitor *ALL drive letters with the prerequisite that an **Update List by Scan** was performed at the endpoint. The Enable Matching option will leverage the drive letter instances discovered.



Agent Procedure

Agent Procedure checks for each drive letter for each alarm instance. Without an **Update List by Scan** the LogObject name will be blank. It will cause the logic to check for the drive letter to fail.

```
1   getVariable ("Agent Working Directory Path", " ", "wkdir", "All Windows Operating Systems", "Halt on Fail")
2   getVariable ("Constant Value", "#ln#", "global:logobject", "All Windows Operating Systems", "Halt on Fail")
3   getVariable ("Constant Value", "#lo#", "global:logobjecttype", "All Windows Operating Systems", "Halt on Fail")
4   getVariable ("Constant Value", "#mn#", "global:monitorsetname", "All Windows Operating Systems", "Halt on Fail")
5   getVariable ("Constant Value", "#lv#", "global:monitorlogvalue", "All Windows Operating Systems", "Halt on Fail")
6   getVariable ("Constant Value", "#id#", "global:idname", "All Windows Operating Systems", "Halt on Fail")
7   getVariable ("Constant Value", "#ao#", "global:alarmoperator", "All Windows Operating Systems", "Halt on Fail")
8   getVariable ("Constant Value", "#av#", "global:alarmthreshold", "All Windows Operating Systems", "Halt on Fail")
9   getVariable ("Constant Value", "#at#", "global:attime", "All Windows Operating Systems", "Halt on Fail")
10  getVariable ("Constant Value", "#body#", "global:alarmbody", "All Windows Operating Systems", "Halt on Fail")
11  getVariable ("Constant Value", "#subject#", "global:alarmsubject", "All Windows Operating Systems", "Halt on Fail")
12  getVariable ("Constant Value", "none", "global:driveletter", "All Operating Systems", "Halt on Fail")
13  if checkVar ("#global:logobject#") Contains "LogicalDisk % Free Space C:"
14      getVariable ("Constant Value", "C:", "global:driveletter", "All Operating Systems", "Halt on Fail")
15  if checkVar ("#global:logobject#") Contains "LogicalDisk % Free Space D:"
16      getVariable ("Constant Value", "D:", "global:driveletter", "All Operating Systems", "Halt on Fail")
17  if checkVar ("#global:logobject#") Contains "LogicalDisk % Free Space E:"
18      getVariable ("Constant Value", "E:", "global:driveletter", "All Operating Systems", "Halt on Fail")
19  if checkVar ("#global:logobject#") Contains "LogicalDisk % Free Space F:"
20      getVariable ("Constant Value", "F:", "global:driveletter", "All Operating Systems", "Halt on Fail")
21  if checkVar ("#global:logobject#") Contains "LogicalDisk % Free Space G:"
22      getVariable ("Constant Value", "G:", "global:driveletter", "All Operating Systems", "Halt on Fail")
23  if checkVar ("#global:logobject#") Contains "LogicalDisk % Free Space H:"
24      getVariable ("Constant Value", "H:", "global:driveletter", "All Operating Systems", "Halt on Fail")
25  if checkVar ("#global:logobject#") Contains "LogicalDisk % Free Space I:"
26      getVariable ("Constant Value", "I:", "global:driveletter", "All Operating Systems", "Halt on Fail")
27  if checkVar ("#global:logobject#") Contains "LogicalDisk % Free Space J:"
28      getVariable ("Constant Value", "J:", "global:driveletter", "All Operating Systems", "Halt on Fail")
29  writeProcedureLogEntry ("Check #global:logobject# - wmic logicaldisk get c...", "All Operating Systems", "Halt on Fail")
30  executeShellCommand ("wmic logicaldisk get caption, description, volume... ", "Execute as System", "All Windows Operating Systems", "Halt on Fail")
31  getVariable ("File Content", "#wkdir#\disktestresult.txt", "global:results", "All Windows Operating Systems", "Halt on Fail")
32  if checkVar ("#global:results#") Contains "Local Fixed Disk"
33      if checkVar ("#global:results#") Contains "Recovery"
34          writeProcedureLogEntry ("PS-Alert #global:attime# on #global:idname#, #... ", "All Windows Operating Systems", "Halt on Fail")
35      else
36          getVariable ("Constant Value", "#global:logobject# Done Via AP #global:monitor... ", "alertSubject", "All Operating Systems", "Halt on Fail")
37          getVariable ("Constant Value", "Done Via AP #global:monitorsetname# #global:a... ", "alertBody", "All Operating Systems", "Halt on Fail")
38          getVariable ("Constant Value", "YES", "alertGenerateTicket", "All Operating Systems", "Halt on Fail")
39          sendAlert ("All Windows Operating Systems", "Halt on Fail")
40  else
41      writeProcedureLogEntry ("PS-Alert #global:attime# on #global:idname#, #... ", "All Windows Operating Systems", "Halt on Fail")
```

To add more drive letters to the agent procedure, repeat the line 13 and 14 replacing the C: with the next letter.