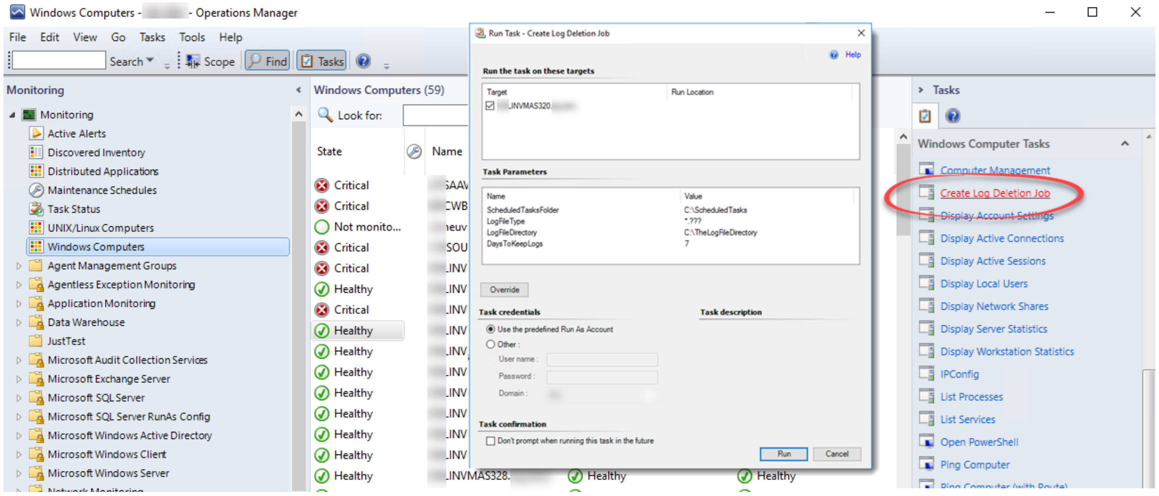
SCOM – Agent Task - Create Log Deletion Job

Summary

Create Log Deletion Job is a SCOM – Agent Task which offers the creation of a scheduled task that deletes log files older than N days on the monitored computer. It works on SCOM 2012 R2 and later.



Run Task - Create Log Deletion Job

Run the task on these targets

Target	Run Location
☑ [redacted]INVMAS320.[redacted]	

Task Parameters

Name	Value
ScheduledTasksFolder	C:\ScheduledTasks
LogFileType	*.???
LogFileDirectory	C:\TheLogFileDirectory
DaysToKeepLogs	7

Override

Task credentials

● Use the predefined Run As Account
○ Other :
User name :
Password :
Domain :

Task description

Task confirmation

□ Don't prompt when running this task in the future

Run Cancel

Table of Contents

Introduction

Many applications and services which run on a server are creating logs. Some are doing it by default and some others are configured to do so by an Administrator or Developer.

To prevent that the logs, occupy the complete disk space either the concerned developer implemented a smart log handling which rotates the files or the Administrator added a scheduled script to do so.

This Management Pack provides a convenient way to create a log deletion job directly out of the SCOM console.

Realization

SCOM offers Agent Tasks to run any command line tool or VBScript directly from the Management Console on a computer that is monitored via agent.

In this case PowerShell is used to run the required code that creates a scheduled task and a PowerShell script to do the cleanup.

The Agent Task

The task itself is realized through a custom module containing a WriteActionModuleType "Windows.Computer.AgentTasks.CreateLogDeletionJob.WriteAction" and a Task "Windows.Comptuer.AgentTasks.CreateLogDeletionJob.Task" which is targeting Windows Computer. – Thus the task is available to all objects based on the Windows Computer class.

The Script

The following script is part of the management pack and launched by the agent task on the monitored computer. It receives the parameter directly from the SCOM console.

Errors and success messages are made visible to the administrator in the SCOM console and in the Operations Manager log file.

```
param($LogFileDirectory,$LogFileType,$DaysToKeepLogs,$ScheduledTasksFolder)


$api = New-Object -ComObject 'MOM.ScriptAPI'
$api.LogScriptEvent('CreateLogDeletionJob.ps1',4000,4,"Script runs. Parameters:
LogFileDirectory $($LogFileDirectory), LogFileType: $($LogFileType) DaysToKeepLogs
$($DaysToKeepLogs) and scheduled task folder $($scheduledTasksFolder)")

Write-Verbose -Message "CreateLogDeletionJob.ps1 with these parameters: LogFileDirectory
$($LogFileDirectory), LogFileType: $($LogFileType) DaysToKeepLogs $($DaysToKeepLogs) and
scheduled task folder $($scheduledTasksFolder)"

$ComputerName           = $env:COMPUTERNAME

$LogFileDirectoryClean = $LogFileDirectory      -Replace('\\','-')
$LogFileDirectoryClean = $LogFileDirectoryClean -Replace(':','')

$scheduledTasksFolder  = $scheduledTasksFolder -replace([char]34,'')
$scheduledTasksFolder  = $scheduledTasksFolder -replace("`"",'')
$taskName              = "Auto-Log-Dir-
Cleaner_for_$($LogFileDirectoryClean)_on_$($ComputerName)"
$taskName              = $taskName -replace '\s',''
$scriptFileName        = $taskName + '.ps1'
$scriptPath            = Join-Path -Path $scheduledTasksFolder -ChildPath $scriptFileName


if ($DaysToKeepLogs -notMatch '\d' -or $DaysToKeepLogs -gt 0) {
        $daysToKeepLogs = 7
        $msg = 'Script warning. DayToKeepLogs not defined or not matching a number.
Defaulting to 7 Days.'
        $api.LogScriptEvent('CreateLogDeletionJob.ps1',4000,2,$msg)
}

if ($scheduledTasksFolder -eq $null) {
        $scheduledTasksFolder = 'C:\ScheduledTasks'
} else {
        $msg = 'Script warning. ScheduledTasksFolder not defined. Defaulting to
C:\ScheduledTasks'
        $api.LogScriptEvent('CreateLogDeletionJob.ps1',4000,2,$msg)
        Write-Warning -Message $msg
}

if ($LogFileDirectory -match 'TheLogFileDirectory') {
        $msg =  'CreateLogDeletionJobs.ps1 - Script Error. LogFileDirectory not defined.
Script ends.'
        $api.LogScriptEvent('CreateLogDeletionJob.ps1',4000,1,$msg)
        Write-Warning -Message $msg
        Exit
}

if ($LogFileType -match '\?\?\?') {
        $msg = 'Script Error. LogFileType not defined. Script ends.'
        $api.LogScriptEvent('CreateLogDeletionJob.ps1',4000,1,$msg)
        Write-Warning -Message $msg
        Exit
}


Function Write-LogDirCleanScript {
        param(
                [string]$scheduledTasksFolder,
                [string]$LogFileDirectory,
                [int]$DaysToKeepLogs,
                [string]$LogFileType,
                [string]$scriptPath
        )

        if (Test-Path -Path $scheduledTasksFolder) {
                $foo = 'folder exists, no action requried'
        } else {
                & mkdir $scheduledTasksFolder
        }

        if (Test-Path -Path $LogFileDirectory) {
                $foo = 'folder exists, no action requried'
        } else {
                $msg = "Script function (Write-LogDirCleanScript, scriptPath:
$($scriptPath)) failed. LogFileDirectory not found $($LogFileDirectory)"
                Write-Warning -Message $msg
                $api.LogScriptEvent('CreateLogDeletionJob.ps1',4001,1,$msg)
                Exit
        }

        if ($LogFileType -notMatch '\*\.[a-zA-Z0-9]{3,}') {
                $LogFileType = '*.' + $LogFileType
                if ($LogFileType -notMatch '\*\.[a-zA-Z0-9]{3,}') {
                        $msg = "Script function (Write-LogDirCleanScript, scriptPath:
$($scriptPath)) failed. LogFileType: $($LogFileType) seems to be not correct."
                        Write-Warning -Message $msg
                        $api.LogScriptEvent('CreateLogDeletionJob.ps1',4001,1,$msg)
                        Exit
                }
        }

$fileContent = @"
Get-ChildItem -Path `"${LogFileDirectory}`" -Include ${LogFileType} -ErrorAction
SilentlyContinue | Where-Object { ((Get-Date) - `$_.LastWriteTime).days -gt
${DaysToKeepLogs} } | Remove-Item -Force
"@

        $fileContent | Set-Content -Path $scriptPath -Force

        if ($error) {
                $msg = "Script function (Write-LogDirCleanScript, scriptPath:
$($scriptPath)) failed. $($error)"
                $api.LogScriptEvent('CreateLogDeletionJob.ps1',4001,1,$msg)
                Write-Warning -Message $msg
        } else {
                $msg = "Script: $($scriptPath) successfully created"
                Write-Verbose -Message $msg
        }

} #End Function Write-LogDirCleanScript


Function Invoke-ScheduledTaskCreation {

        param(
                [string]$ComputerName,
                [string]$taskName
        )

        $currentTasks = & SCHTASKS /Query /FO CSV
        $currentTasks = $currentTasks -replace "`"TaskName`",`"Next Run
Time`",`"Status`"",""
        $currTasks    = ConvertFrom-Csv -InputObject $currentTasks -Header ("TaskName",
"Next Run Time", "Status")
        $foundTasks   = $currTasks | Where-Object {$_.TaskName -match 'Auto-Log-Dir-
Cleaner'}
```

```
        if ($foundTasks) {
                $msg = "Script function (Invoke-ScheduledTaskCreation) foundTask:
$($foundTasks.ToString()) already. No action required."
                Write-Verbose -Message $msg
                $api.LogScriptEvent('CreateLogDeletionJob.ps1',4002,4,$msg)
        } else {
                $taskRunFile          =
"C:\WINDOWS\system32\WindowsPowerShell\v1.0\powershell.exe -NoLogo -NonInteractive -File
$($scriptPath)"
                $taskStartTimeOffset = Get-Random -Minimum 1 -Maximum 10
                $taskStartTime        = (Get-Date).AddMinutes($taskStartTimeOffset) | Get-
date -Format 'HH:mm'

                $taskSchedule         = 'DAILY'
                & SCHTASKS /Create /SC $($taskSchedule) /RU `"NT AUTHORITY\SYSTEM`" /TN
$($taskName) /TR $($taskRunFile) /ST $($taskStartTime)
        }

        if ($error) {
                $msg = "Sript function (Invoke-ScheduledTaskCreation) Failure during task
creation! $($error)"
                $api.LogScriptEvent('CreateLogDeletionJob.ps1',4002,1,$msg)
                Write-Warning -Message $msg
        } else {
                $msg = "Scheduled Tasks: $($taskName) successfully created"
                Write-Verbose -Message $msg
        }

} #End Function Invoke-ScheduledTaskCreation


$logDirCleanScriptParams    = @{
        'scheduledTasksFolder' = $ScheduledTasksFolder
        'LogFileDirectory'     = $LogFileDirectory
        'daysToKeepLogs'       = $DaysToKeepLogs
        'LogFileType'          = $LogFileType
        'scriptPath'           = $scriptPath
}

Write-LogDirCleanScript @logDirCleanScriptParams


$taskCreationParams = @{
        'ComputerName'  = $ComputerName
        'taskName'      = $taskName
        'scriptPath'    = $scriptPath
}

Invoke-ScheduledTaskCreation @taskCreationParams
```

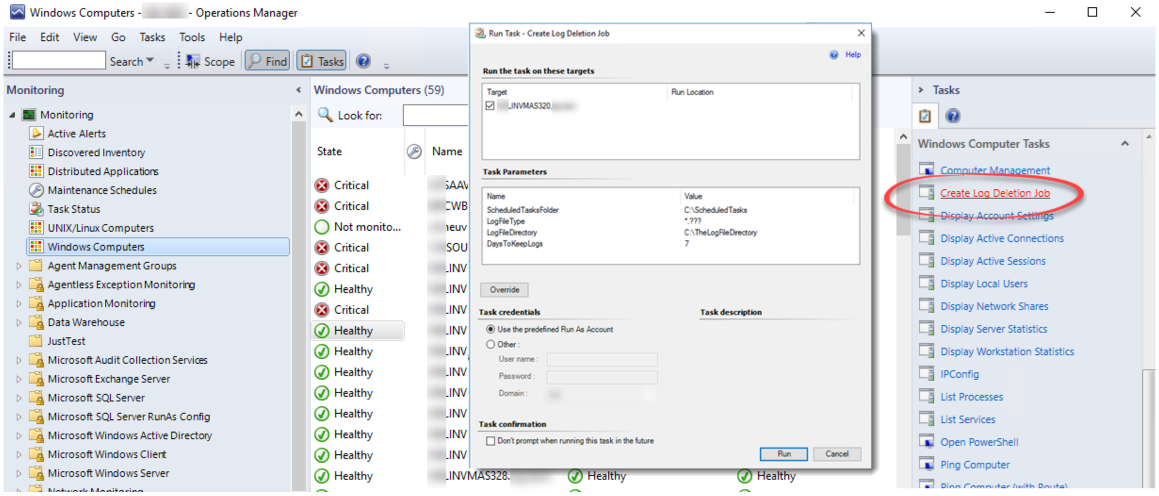
Usage

- Download and Import the Management Pack from [URL}.

- In the SCOM Console, Monitoring pane, click on Windows Computer for instance.

- On the right, in the 'Task' pane below Windows Computer Tasks you can find Create Log Deletion Job

- Select a computer object and click on 'Create Log Deletion Job'.



Run Task - Create Log Deletion Job

Run the task on these targets

Target	Run Location
☑ INVMAS320.	

Task Parameters

Name	Value
ScheduledTasksFolder	C:\ScheduledTasks
LogFileType	*.???
LogFileDirectory	C:\TheLogFileDirectory
DaysToKeepLogs	7

Task credentials

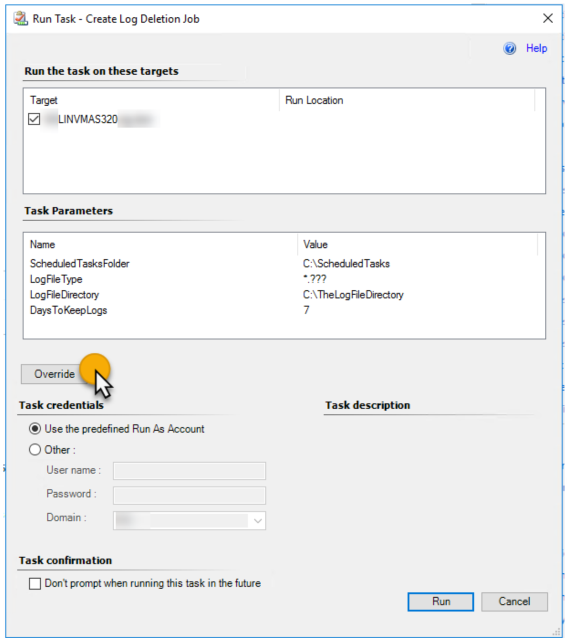
● Use the predefined Run As Account

○ Other :

Task confirmation

□ Don't prompt when running this task in the future

- Click on the Override button to change the values according your requirement. E.g. the logs are in 'C:\abc', the extension of the logfile is *.abc and you like to keep only logs your than 8 days:



Run Task - Create Log Deletion Job

Run the task on these targets

Target	Run Location
☑ LINVMAS320	

Task Parameters

Name	Value
ScheduledTasksFolder	C:\ScheduledTasks
LogFileType	*.???
LogFileDirectory	C:\TheLogFileDirectory
DaysToKeepLogs	7

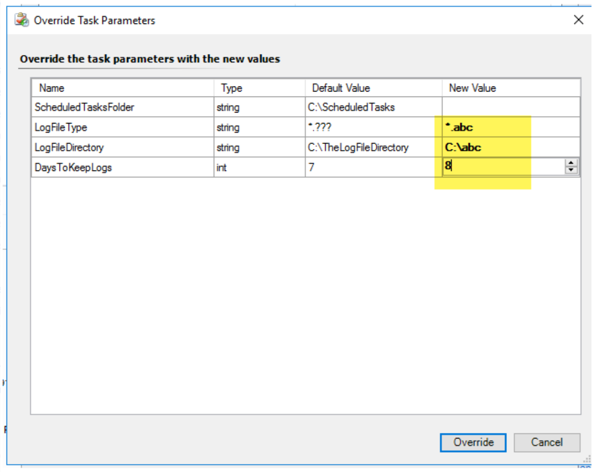
Task credentials

● Use the predefined Run As Account

○ Other :

Task confirmation

□ Don't prompt when running this task in the future

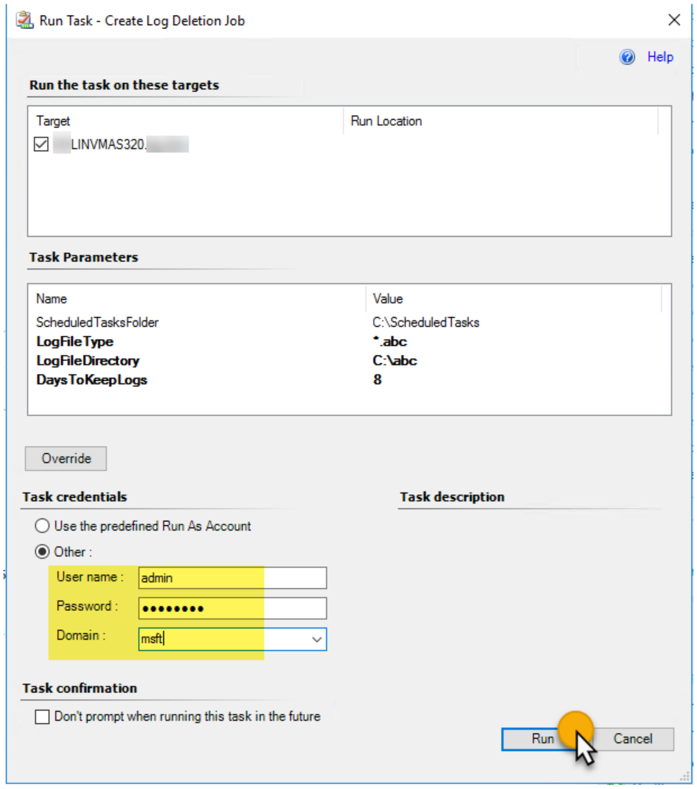


Override Task Parameters

Override the task parameters with the new values

Name	Type	Default Value	New Value
ScheduledTasksFolder	string	C:\ScheduledTasks	
LogFileType	string	*.???	*.abc
LogFileDirectory	string	C:\TheLogFileDirectory	C:\abc
DaysToKeepLogs	int	7	8

- Set the credentials that should be used to run the task on the remote machine. – The account need to have administrative permissions on the target machine. Confirm with 'Run'.



Run Task - Create Log Deletion Job

Run the task on these targets

Target	Run Location
☑ [redacted] LINVMAS320.[redacted]	

Task Parameters

Name	Value
ScheduledTasksFolder	C:\ScheduledTasks
LogFileType	***.abc**
LogFileDirectory	**C:\abc**
DaysToKeepLogs	**8**

Task credentials

○ Use the predefined Run As Account

● Other :

User name : admin

Password : ••••••••

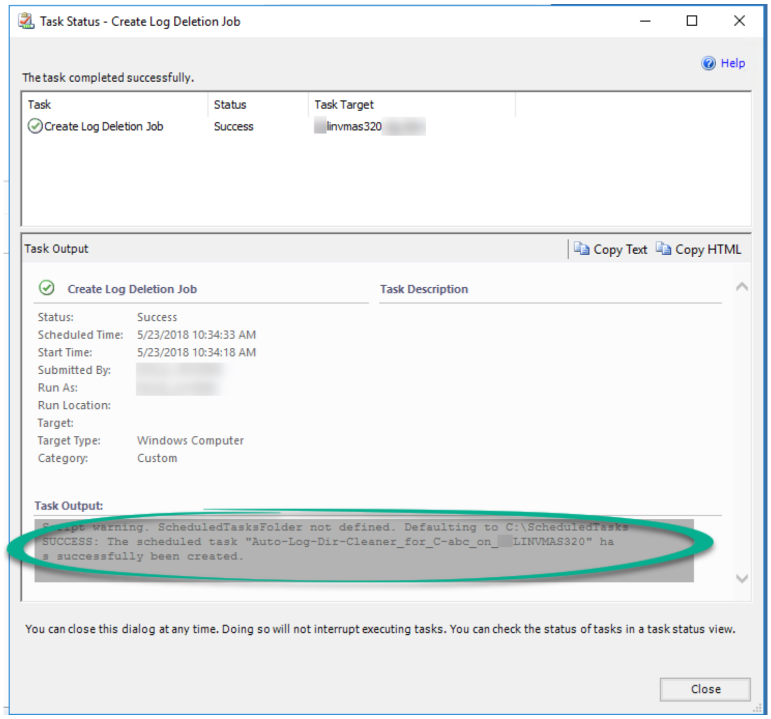
Domain : msft

Task description

Task confirmation

☐ Don't prompt when running this task in the future

- If all goes fine a success message will confirm that the task job was created:



The task completed successfully.

Task	Status	Task Target
✓ Create Log Deletion Job	Success	[redacted] linvmas320 [redacted]

✓ Create Log Deletion Job

Task Description

Status: Success
Scheduled Time: 5/23/2018 10:34:33 AM
Start Time: 5/23/2018 10:34:18 AM
Submitted By: [redacted]
Run As: [redacted]
Run Location:
Target:
Target Type: Windows Computer
Category: Custom

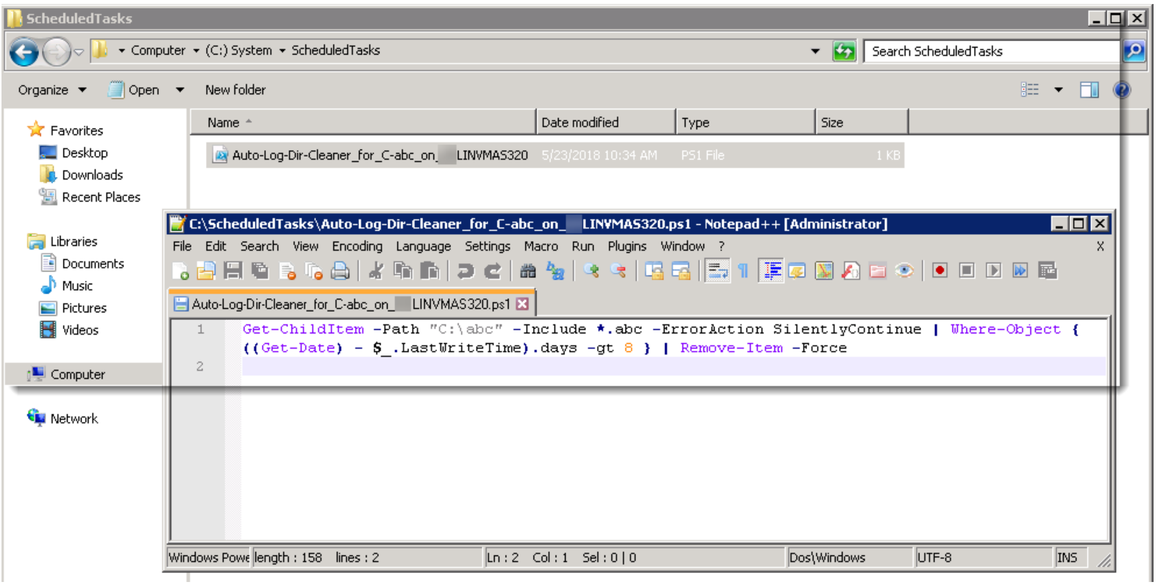
Task Output:

```
Script warning. ScheduledTasksFolder not defined. Defaulting to C:\ScheduledTasks
SUCCESS: The scheduled task "Auto-Log-Dir-Cleaner_for_C-abc_on_[redacted]LINVMAS320" ha
s successfully been created.
```

You can close this dialog at any time. Doing so will not interrupt executing tasks. You can check the status of tasks in a task status view.

Checking the result

On the target machine then, a PowerShell script will be available in 'C:\ScheduledTasks' if the value was not changed to something different.



ScheduledTasks

Computer ▸ (C:) System ▸ ScheduledTasks

Name	Date modified	Type	Size
Auto-Log-Dir-Cleaner_for_C-abc_on_[redacted]LINVMAS320	5/23/2018 10:34 AM	PS1 File	1 KB

C:\ScheduledTasks\Auto-Log-Dir-Cleaner_for_C-abc_on_[redacted]LINVMAS320.ps1 - Notepad++ [Administrator]

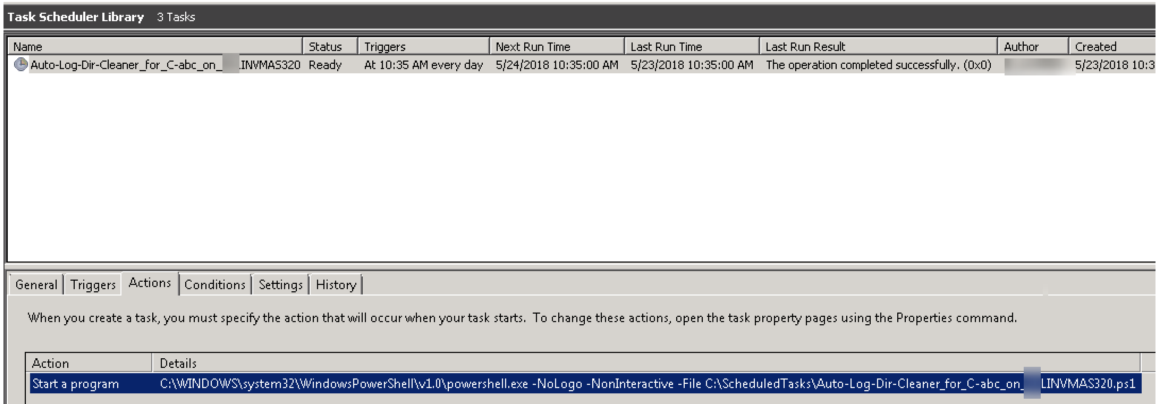
Auto-Log-Dir-Cleaner_for_C-abc_on_[redacted]LINVMAS320.ps1

```
Get-ChildItem -Path "C:\abc" -Include *.abc -ErrorAction SilentlyContinue | Where-Object {
((Get-Date) - $_.LastWriteTime).days -gt 8 } | Remove-Item -Force

```

Windows Powe length : 158 lines : 2 Ln : 2 Col : 1 Sel : 0 | 0 Dos\Windows UTF-8 INS

A scheduled task is created which runs the script daily.



Task Scheduler Library 3 Tasks

Name	Status	Triggers	Next Run Time	Last Run Time	Last Run Result	Author	Created
Auto-Log-Dir-Cleaner_for_C-abc_on_[redacted]INVMAS320	Ready	At 10:35 AM every day	5/24/2018 10:35:00 AM	5/23/2018 10:35:00 AM	The operation completed successfully. (0x0)	[redacted]	5/23/2018 10:3

When you create a task, you must specify the action that will occur when your task starts. To change these actions, open the task property pages using the Properties command.

Action	Details
Start a program	C:\WINDOWS\system32\WindowsPowerShell\v1.0\powershell.exe -NoLogo -NonInteractive -File C:\ScheduledTasks\Auto-Log-Dir-Cleaner_for_C-abc_on_[redacted]LINVMAS320.ps1

Note: The scheduled task runs as 'SYSTEM'.