* ***Doc for adding extents alert***

In order to place the Variable extents alerts on server please follow the following steps

1. Create folder as C:\Scripts\Monitoring and create the following two scripts in same folder **CriticalExtentsChecker.bat** and include the following content

REM critical extents checker script

C:

cd C:\Scripts\Monitoring

C:\Progress\OpenEdge11.7\bin\prowin.exe -db F:\AppriseDB\8.0.0\Live\apprise -p F:\Scripts\Monitoring\check\_variable\_extent.p -b > C:\replication\monitoring\apprise.txt

C:\Progress\OpenEdge11.7\bin\prowin.exe -db F:\AppriseDB\8.0.0\Live\custom -p F:\Scripts\Monitoring\check\_variable\_extent.p -b > C:\replication\monitoring\custom.txt

1. Create one more PowerShell script with the following content and give name **CriticalExtentsMonitoring.ps1**

#--------- script to alert about Critical Variable Extents ---------#

[Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]::Tls12

#-------- define variables ---------#

select-string -Path "C:\replication\monitoring\Apprise.txt" -Pattern "true" | select Line | out-file C:\replication\monitoring\CriticalApprise.txt

select-string -Path "C:\replication\monitoring\Custom.txt" -Pattern "true" | select Line | out-file C:\replication\monitoring\CriticalCustom.txt

$sortedfile = Get-Content -Path 'C:\replication\monitoring\CriticalApprise.txt'

$sortedfile1 = Get-Content -Path 'C:\replication\monitoring\CriticalCustom.txt'

$exec = ([regex]::Matches((Get-content "C:\replication\monitoring\criticalApprise.txt"), "\d+")).Value

$exec1 = ([regex]::Matches((Get-content "C:\replication\monitoring\criticalCustom.txt"), "\d+")).Value

$file = (Get-ChildItem C:\replication\monitoring\Apprise.txt).BaseName

$file1 = (Get-ChildItem C:\replication\monitoring\Custom.txt).BaseName

#--------- send email for action performed ---------#

if ($sortedfile -cmatch ("true") )

{

$output = "OOly02 has Critical Extents "+ $exec +" for their Live $file database. "

$userName = 'noreply-apprise@aptean.com'

$password = 'Winter@123'

[SecureString]$securepassword = $password | ConvertTo-SecureString -AsPlainText -Force

$credential = New-Object System.Management.Automation.PSCredential -ArgumentList $username, $securepassword

Send-MailMessage -SmtpServer smtp.office365.com -Port 587 -UseSsl -From noreply-apprise@aptean.com -To ApteanSRE-Jedi@aptean.com -Subject 'CRITICAL - Extents are critical' -Body $output -Credential $credential

}

if ($sortedfile -cmatch ("true") )

{

$output = "OOly02 has Critical Extents "+ $exec1 +" for their Live $file1 database. "

$userName = 'noreply-apprise@aptean.com'

$password = 'Winter@123'

[SecureString]$securepassword = $password | ConvertTo-SecureString -AsPlainText -Force

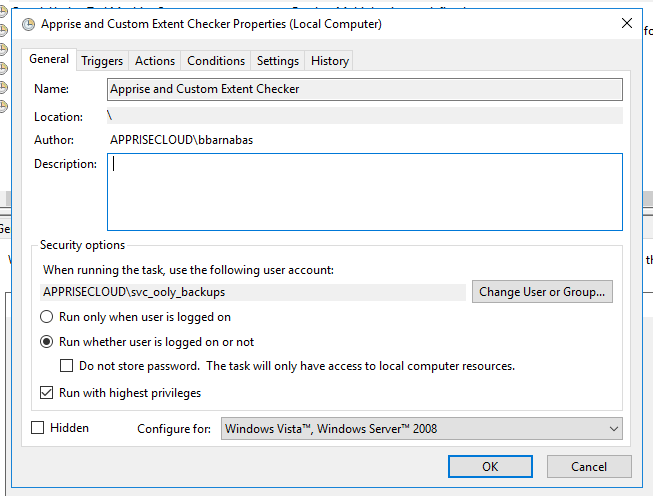
$credential = New-Object System.Management.Automation.PSCredential -ArgumentList $username, $securepassword

Send-MailMessage -SmtpServer smtp.office365.com -Port 587 -UseSsl -From noreply-apprise@aptean.com -To ApteanSRE-Jedi@aptean.com -Subject 'CRITICAL - Extents are critical' -Body $output -Credential $credential

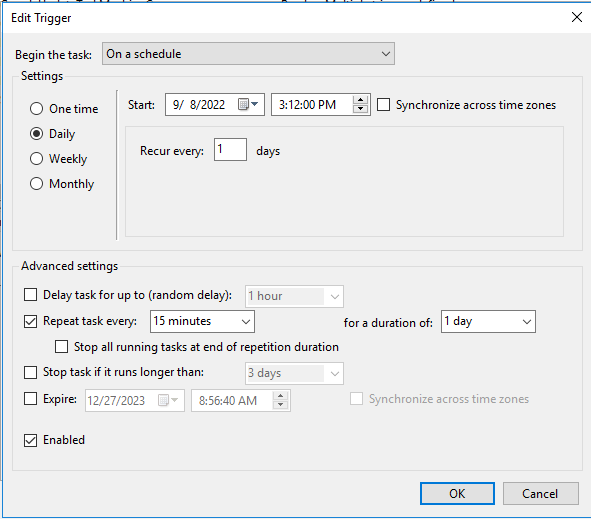
}

1. Once scripts has been created, please create job in task scheduler for executing batch script with name CriticalExtentsChecker. Use the backup service account to run the job as this account has the run batch as job permissions

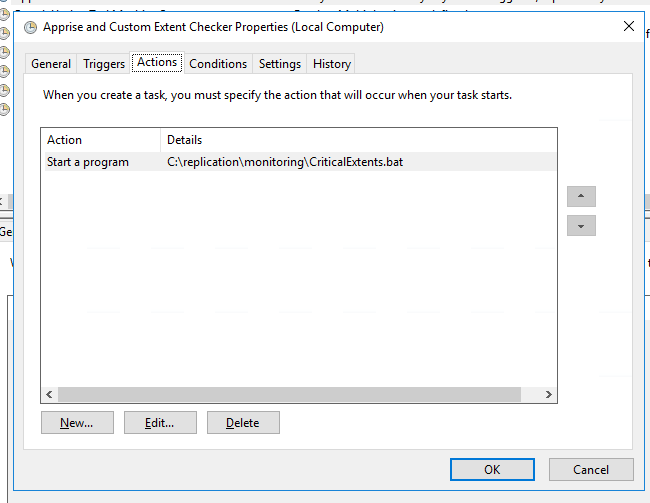
**General Settings -**



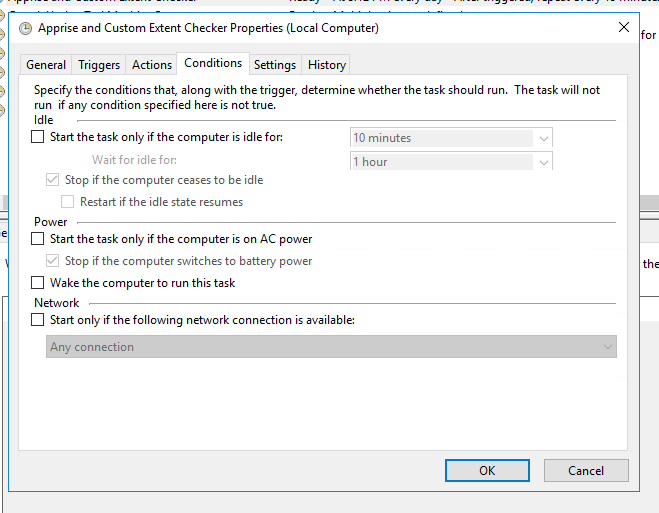
**Edit Trigger - trigger should be every 15 minutes**



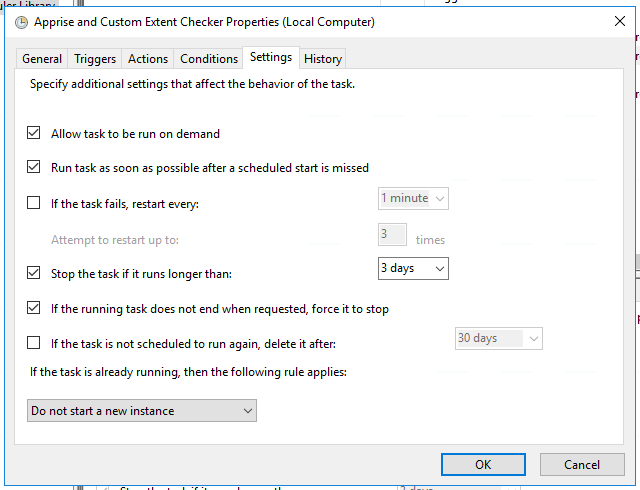
**Actions - Select the batch script to run as job**



**Condition - unselect the power related checkbox**



**Settings - select the following check boxes**



Use the same steps to schedule job for PowerShell alerting script. Also, add the trigger the job to be executed after every 1 hour.