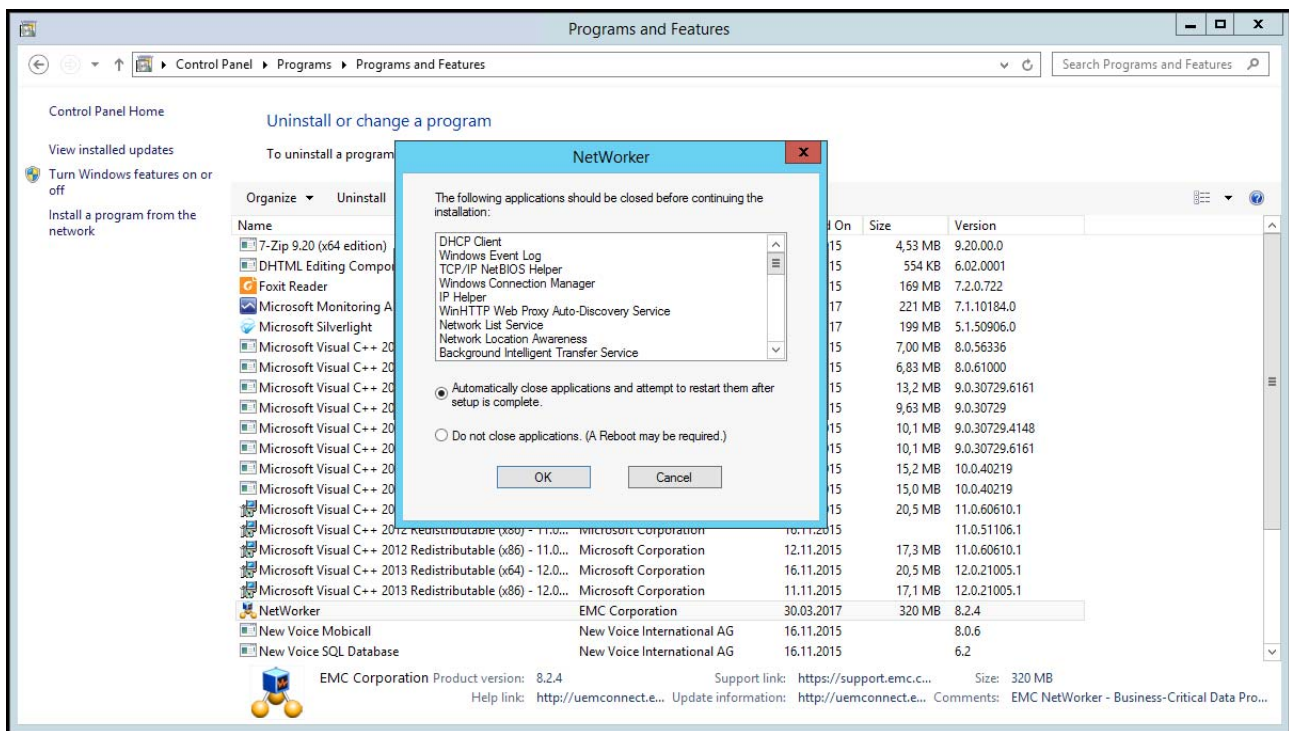


Ein Client Push Update startet andere Services bzw. sogar den Server (den NetWorker Client)

Das darf natürlich nicht passieren - und es ist mir bislang auch nur ein einziges Mal vorgekommen. Es handelt sich genau um dieses Szenario, das Ihnen aber auch nur bei einem manuellen Update auffällt (einer von mehr als 500 Servern/NetWorker Clients):



Der NetWorker benötigt diese Restarts nicht - Sie können mit *Do not close applications* fortfahren und den Update trotzdem durchführen.

Bei einem NetWorker Client-Push Update geht das Installationsprogramm (Windows `msiexec.exe`) einfach weiter vor und startet die Services durch. Das bedeutet zumindest einen 'kurzen' Unterbruch für Ihre Applikation.

Ich habe versucht, das Problem nachzustellen und zu untersuchen; dies ist mir jedoch bislang nicht gelungen. Bei Dell/EMC war ein Service Request bzw. ein *Request for Enhancement/Request for Improvement* offen, damit der NetWorker eine solche Situation zu Anfang der Installation zumindest prüft und den Upgrade für diesen Client abbricht. Mit Stand 04/2019 wurde dies allerdings wieder verworfen, da ...

- seitens des Herstellers (also Dell/EMC) keine logische Notwendigkeit besteht - der NetWorker trägt eine solche Abhängigkeit nicht selbst ein
- der Aufwand für die wenigen Problemfälle zu hoch ist.



Momentan kann ich Sie deshalb nur warnen und Sie darauf hinweisen, daß Sie potentiell kritische Clients besser manuell upgraden.

Allerdings habe ich mittlerweile im Internet doch ein PowerShell Programm gefunden, mit dessen Hilfe Sie vor einem Update die Service-Abhängigkeit überprüfen können.

Hier meine adaptierte PowerShell Programm:

```

# #####
# ##### Allgemeine Funktionen definieren #####
# #####
#
# #####
# ##### touch - erstellen einer leeren Datei #####
# #####
#
function touch
{
    set-content -path ($args[0]) -value ($null)
}
#
# #####
# ##### pause - 'Breakpoint' Funktion #####
# #####
#
function pause ($Message = "Weiter mit einer beliebigen Taste ...")
{
    Write-Host -NoNewLine $Message
    $null = $Host.UI.RawUI.ReadKey("NoEcho,IncludeKeyDown") | out-null
}
#
#
#####
#
# #####
# ##### 1. Globale Variable definieren #####
# #####
#
Function globale_variable_definieren
{
    $global:out_path = "D:\temp\"
    $global:nsr_path = "D:\temp\"
}
#
# #####
# ##### 2. Umgebung setzen #####
# #####
#
function umgebung_setzen
{
    $culture = New-Object //
    System.Globalization.CultureInfo("en-US")
    $host.ui.RawUI.WindowTitle = "Abhaengigie_nw_client_services_ermitteln.ps1"
}
#
#
.....

```

```

.....

# #####
# ##### 3. Aktuellen Zeitstempel ermitteln #####
# #####
#
function aktuelle_zeit_ermitteln
{
    $datum          = get-date -format G
    $datum          = (get-date).addminutes(-1)
    $global:jahr    = "{0:yyyy}" -f $datum
    $global:monat   = "{0:MM}" -f $datum
    $tag            = "{0:dd}" -f $datum
    $stunde         = "{0:HH}" -f $datum # "hh" = 12Std., "HH" = 24Std.
    $minute         = "{0:mm}" -f $datum
    $sekunde        = "{0:ss}" -f $datum
    $timestamp      = "$monat" + "/" + "$tag" + "/" + "$jahr" + " " `
        + "$stunde" + ":" + "$minute" + ":" + "$sekunde"
}
#
# #####
# ##### 4. Dependent Services ermitteln #####
# #####
#
function dependent_services_ermitteln
{
    $service_name   = "nsrexecd"
    $datei_ein      = $global:out_path + "hostnames.txt"
    $datei_aus      = $global:out_path + "dependent_services.txt"
    touch $datei_aus

    $alle_hostnames = Get-Content $datei_ein | select-object -skip 0

    foreach ($hostname in $alle_hostnames)
    {
        $zeile = $hostname
        $ds     = get-service $service_name -dependentservices `
            -computername $hostname
        $zeile = $zeile + "`t`t" + $ds.name
        out-file -FilePath $datei_aus -Force -Encoding ascii -append `
            -inputobject $zeile
    }
    write-host "`r`n`r`nFertig - Beenden mit einer beliebigen Taste ... `
        `r`n`r`n"
    pause $eingabe
}
#
# #####
# ##### HAUPTPROGRAMM #####
# #####
#
globale_variable_definieren
umgebung_setzen
aktuelle_zeit_ermitteln
dependent_services_ermitteln

```

Das Programm ist nicht perfekt. Hieran sollten Sie denken:

- In Schritt 1 müssen Sie die Pfadnamen höchstwahrscheinlich anpassen.
- Vor dem Lauf müssen Sie zuerst noch die Liste aller zu durchsuchenden NetWorker Clients erstellen (`hostnames.txt`).
- Das Programm gibt Fehler bzw. keine Ausgabe für bestimmte NetWorker Clients, bei...
 - UNIX/Linux Rechnern
 - 'Leeren' NetWorker Client Namen
 - Clients, die angelegt sein müssen, wo aber nie die NetWorker Client Software installiert wurde (z.B. eine vCenter Appliance)
 - Cluster Clients