

Domain Controller Decommission Process

STATIC DOCUMENT INFORMATION

Please update the following information manually:

|  |  |
| --- | --- |
| Document Type | How to |
| Service Portfolio | Windows Server Operations |
| Services | Active Directory and DNS |
| Service Component | Microsoft |
| System | Windows Systems |
| Document Owner | Chris Stradling\Harjot Kandola |
| Next Review Date | June 2020 |
| Keywords (separate with “;”) | AD; DNS; Active Directory |

Dynamic Document Information

The following information will be automatically updated.

|  |  |
| --- | --- |
| Document Name | AD Domain Controller Decom.docx |
| Version | 1.2 – Pmorgan – Updated with DNS logging section  .1 – PMorgan – Updated with ConnectionLogger section  1 – PMorgan – First draft version |
| Author | Peter Morgan and Peter Stapleton |
| Date Created | 05/06/2019 |
| Date Last Modified | 25/10/2019 |

Table of Contents

[Summary 4](#_Toc61609718)

[1 De-Advertise Domain Controller 5](#_Toc61609719)

[1.1 DHCP Considerations and Checks 5](#_Toc61609720)

[1.2 DNS Service Record Removal 7](#_Toc61609721)

[1.3 De-advertise in AD Site 10](#_Toc61609722)

[1.4 Discover and Remove Statically assigned DNS dependencies 12](#_Toc61609723)

[Windows Domain Joined DNS dependency Check 12](#_Toc61609724)

[Non-Windows, or Non-Domain Joined DNS dependency Check 14](#_Toc61609725)

[2 Pro-active Monitoring of De-Advertised Domain Controller – Perfmon Method 19](#_Toc61609726)

[2.1 Configure Performance Monitor 19](#_Toc61609727)

[2.2 Collect and Convert Results 22](#_Toc61609728)

[2.3 Analyse Results 23](#_Toc61609729)

[3 Pro-active Monitoring of De-Advertised Domain Controller – Connection Logger Method 27](#_Toc61609730)

[3.1 Configure Connection Logger 27](#_Toc61609731)

[3.2 Collect Connection Logger results 28](#_Toc61609732)

[3.3 Analyse Connection Logger Results 29](#_Toc61609733)

[Generate IP to name list 29](#_Toc61609734)

[Generate Port number to name list 29](#_Toc61609735)

[Add columns containing Vlookup to ConnectionLog.csv 30](#_Toc61609736)

[4 Reactive Monitoring of De-Advertised Domain Controller 31](#_Toc61609737)

[5 Demote Domain Controller 32](#_Toc61609738)

[5.1 Pre Demotion checks 32](#_Toc61609739)

[Global Catalog Check 32](#_Toc61609740)

[FSMO Role Holder Check 32](#_Toc61609741)

[DNS Server Forwarding Check 32](#_Toc61609742)

[Conditional Forward Zone – Cross Forest Check 33](#_Toc61609743)

[DNS Server For DHCP Clients Check 33](#_Toc61609744)

[5.2 Demotion Processes 33](#_Toc61609745)

[5.2.1 Windows 2008 R2 Steps 33](#_Toc61609746)

[5.2.2 Windows 2012 or later Steps 36](#_Toc61609747)

[5.2.3 Verification Post Check – All Versions 42](#_Toc61609748)

[6 TroubleShooting Domain Controller Demotion 43](#_Toc61609749)

[6.1 Known Issues 43](#_Toc61609750)

[6.2 Clean up metadata using GUI tools 43](#_Toc61609751)

[Using Active Directory Users and Computers 44](#_Toc61609752)

[Using Active Directory Sites and Services 44](#_Toc61609753)

# Summary

The purpose of this document is to assist administrators in the process of safely removing Domain Controllers from Active Directory. This process may be required due to office relocation or operating system roadmap refresh, or there may be many Domain Controllers to be removed\replaced as part of a larger project or effort to upgrade DFL\FFL within a forest. Its main objective is to gracefully remove domain controllers from a domain, but also to expose any dependencies from application servers or similar which have been hard coded to the target domain controller and would be affected by its demotion and decommission. In the case of such a dependency, administrators will be able to advise application/ server owners to take action accordingly.

The process is in 4 stages.

1 Remove dependencies

2 De-Advertise

3 Monitor and remove DNS dependencies

4 Demote

Further details around decom of a domain controller can be found here.

<https://social.technet.microsoft.com/wiki/contents/articles/50925.active-directory-checklist-for-decommissioning-a-domain-controller.aspx#The_Checklist>

# 1 De-Advertise Domain Controller

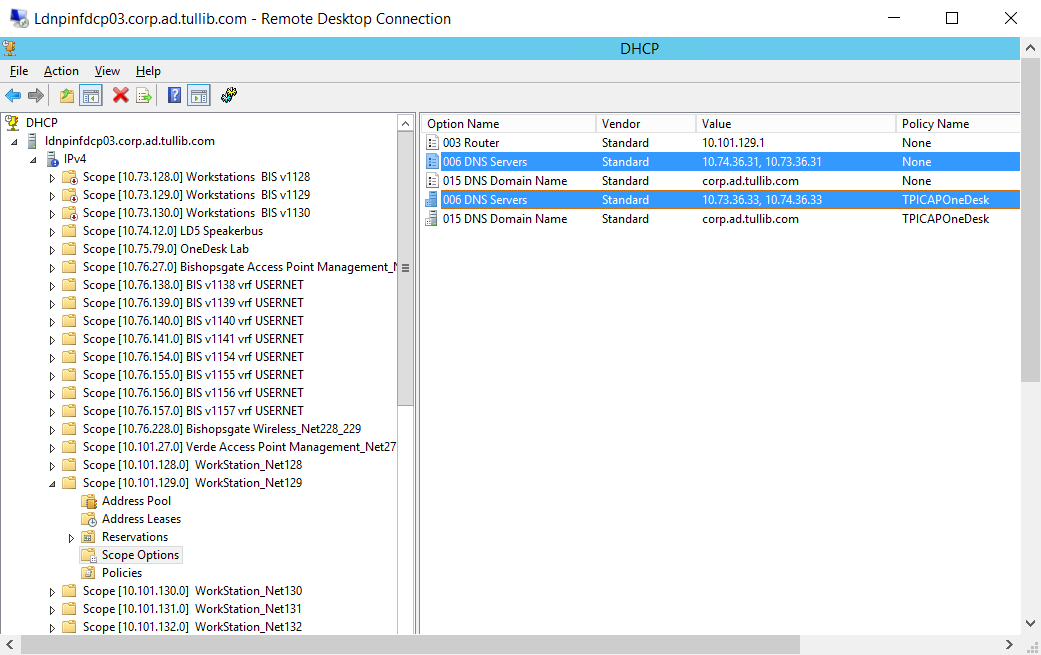
An administrator may or may not decide the de-advertise process is necessary prior to beginning full demotion of a domain controller.

You may decide to use this process if a domain controller has been in production for a long period of time, is running an older version of Windows, or is running in a regional hub AD site (London, Singapore, New Jersey) where large dependent infrastructure\ datacenters are present.

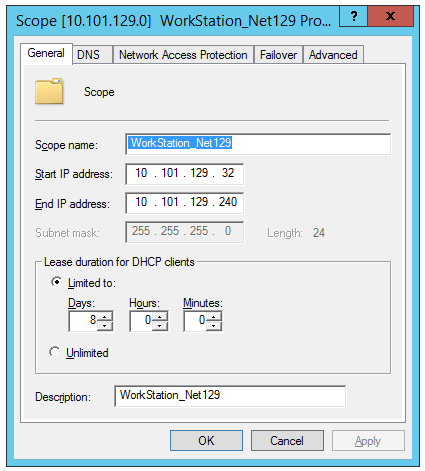
You may decide this process is unnecessary if the DC is running in a branch office with little to no “local” infrastructure is present, it’s only been in production for a short period, or if it’s already running a modern OS. If this is the case, please skip to section 3 “Demote Domain Controller”.

## 1.1 DHCP Considerations and Checks

1. Firstly, make a note of whether or not the Domain Controller is also a DHCP server. If this is true, then it will be OK to demote the server, but de-advertising it may not be effective. Therefore, a determined effort should be made to remove the server's DHCP role and move this to an alternative dedicated DHCP server. This is out of the scope of this document.
2. On each DHCP server in the environment, step through the scopes and make sure that the Domain Controller being decommissioned is not called out within any of the Scope options as a DNS server.

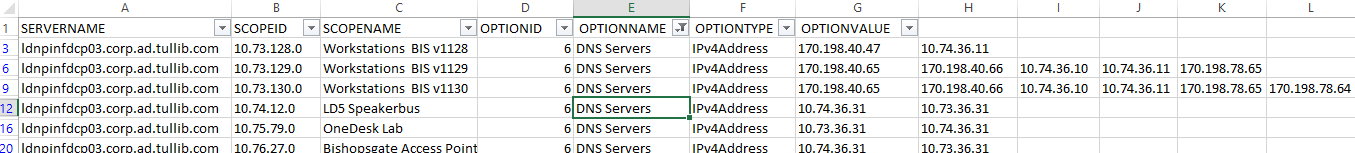


1. Note also the Scope duration that contained the Domain controller. It will be necessary to wait beyond this time for clients to stop communicating with the server for reduced traffic.



Alternatively, since TPICAP has thousands of scopes globally and it may not be possible to check them all for references to the IP address of the domain controller being decommissioned, run the following PowerShell to pull a CSV of all DHCP options in all DHCP scopes on all DHCP servers in a domain. This can then be filtered by server and optionname (DNS Servers) to see all values used across all scopes and all servers in the domain. **Please ensure this is run against ALL DOMAINs, specifically the domain in which the domain controllers being removed reside.**

|  |
| --- |
| $domain = “corp”  $domainsearch = "\*$domain\*"  Add-Content -Value "SERVERNAME,SCOPEID,SCOPENAME,OPTIONID, OPTIONNAME, OPTIONTYPE, OPTIONVALUE" -Path ("C:\temp\" + $domain + "\_DHCP\_Scope\_Options.csv")  $dhcpservers = (Get-ADObject -SearchBase 'cn=configuration,dc=ad,dc=tullib,dc=com' -Filter 'objectclass -eq "dhcpclass" -AND name -ne "dhcproot" -AND name -like $domainsearch').name  #$dhcpservers = "ldnpinfdcp03.corp.ad.tullib.com"  Foreach ($dhcpserver in $dhcpservers) {  $Scopes = Get-DhcpServerv4Scope -ComputerName $dhcpserver | Select-Object ScopeId,Name  foreach ($Scope in $Scopes) {  $Options = Get-DhcpServerv4OptionValue -ComputerName $dhcpserver -ScopeId $Scope.ScopeId.IPAddressToString -All | Sort-Object -Descending -Property OptionId  for ($i = ($Options.Count -1); $i -gt -1; $i--) {  Add-Content -Value "$($dhcpserver),$($Scope.ScopeId.IPAddressToString),$($Scope.Name),$($Options[$i].OptionId),$($Options[$i].Name),$($Options[$i].Type),$($Options[$i].Value -join ',')" -Path "C:\temp\Windows\_Scope\_Options.csv"  }  }  } |

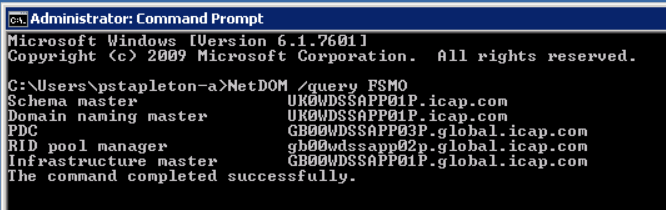


## 1.2 DNS Service Record Removal

This section de-advertises the domain controller for discovery for services like LDAP, Kerberos and NTLM.

1. Logon to the desktop of the target Domain controller and check that it does not hold any FSMO Roles. From a command prompt, type:

NetDOM /query FSMO



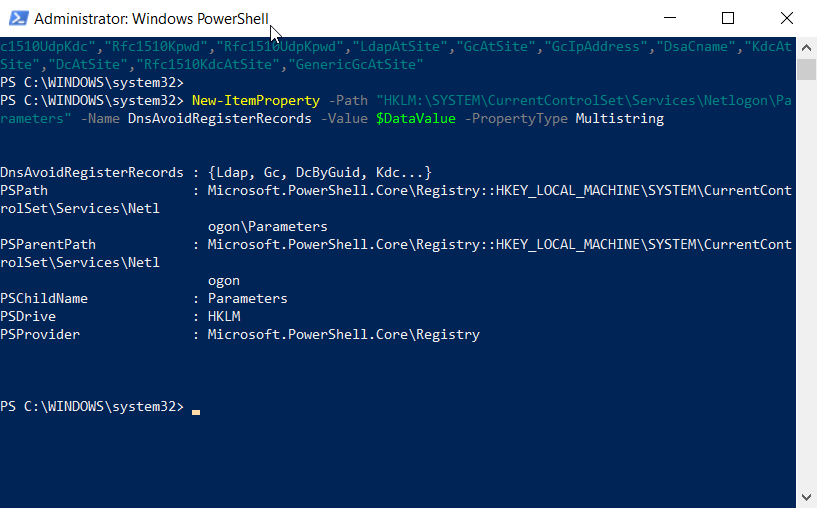
Transfer any roles off of the server if found (this process is beyond the scope of this doc, but can be achieved via ntdsutil or the GUI MMC tools)

1. In this step the action will add a registry key within the Netlogon service to prevent and remove DNS record registration in DNS relevant to this server as a domain controller.

Run the below powershell as shown

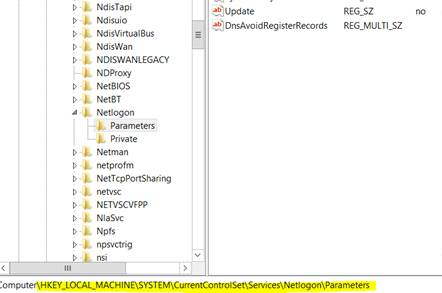
|  |
| --- |
| $DataValue = "Ldap","Gc","DcByGuid","Kdc","Dc","Rfc1510Kdc","GenericGc","Rfc1510UdpKdc","Rfc1510Kpwd","Rfc1510UdpKpwd","LdapAtSite","GcAtSite","GcIpAddress","KdcAtSite","DcAtSite","Rfc1510KdcAtSite","GenericGcAtSite"  New-ItemProperty -Path "HKLM:\SYSTEM\CurrentControlSet\Services\Netlogon\Parameters" -Name DnsAvoidRegisterRecords -Value $DataValue -PropertyType Multistring |

Restart-Service -Name Netlogon



1. In Registry Editor, navigate to the following registry key:

HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\Netlogon\Parameters

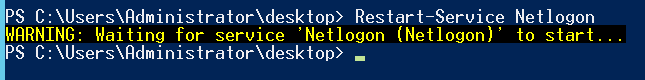


1. Verify the following multistring value (REG\_MULTI\_SZ) value “DnsAvoidRegisterRecord”s has been added with the Values as stated in the PowerShell above. There should be 18 values in the Multi-String.

|  |
| --- |
| Ldap  Gc  DcByGuid  Kdc  Dc  Rfc1510Kdc  GenericGc  Rfc1510UdpKdc  Rfc1510Kpwd  Rfc1510UdpKpwd  LdapAtSite  GcAtSite  GcIpAddress  KdcAtSite  DcAtSite  Rfc1510KdcAtSite  GenericGcAtSite |

1. Now Restart Netlogon on the Domain Controller, once complete wait 15 minutes.

|  |
| --- |
| Restart-Service -Name Netlogon |

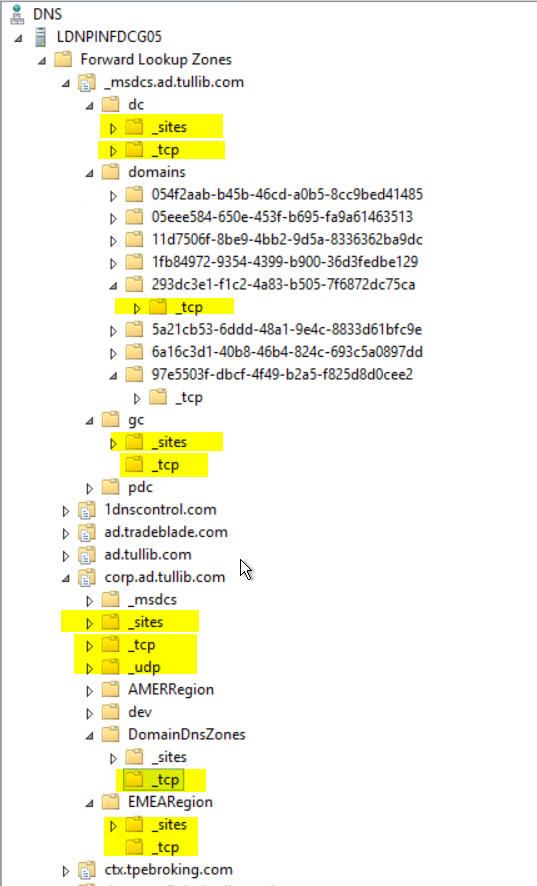
[](https://msdnshared.blob.core.windows.net/media/TNBlogsFS/prod.evol.blogs.technet.com/CommunityServer.Blogs.Components.WeblogFiles/00/00/01/01/77/Capture92.PNG)

1. Verify the above had taken effect by checking DNS. It should show that the Domain controller is now de-advertised by having removed all of it’s records from many areas within DNS under “\_msdcs” zones.

Some example areas are indicated in illustration below. Expand the “Sites” zones and locate the site from which the Domain controller has been contained within.

Note that the actual zones will vary according to the domain and forest of the Domain controller being de-Advertised.

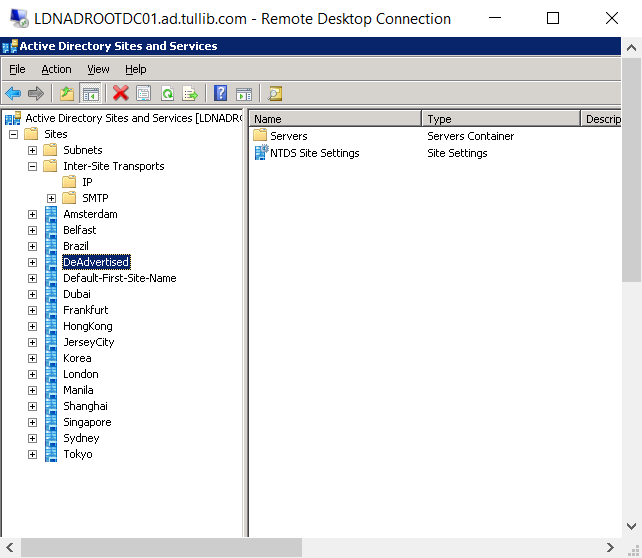
Reference: <https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2003/cc778029(v=ws.10)>



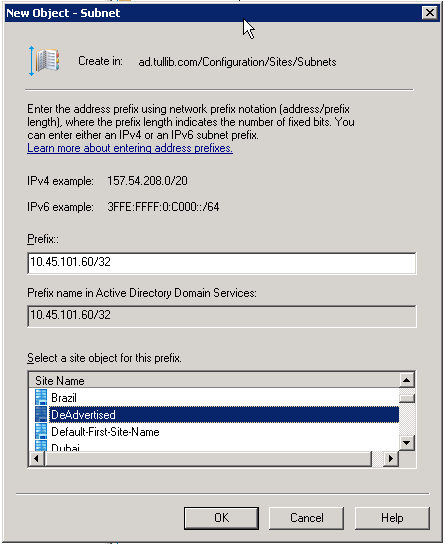
## 1.3 De-advertise in AD Site

A full approach to hiding or de-advertising a Domain Controller would be to move the DC to an unused or “De-Advertised” AD site. This can be done in addition to the removal of DNS SRV records as described above if desired.

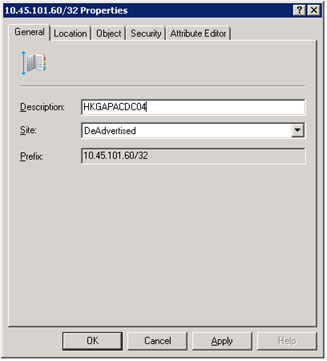
1. The AD site “DeAdvertised” should already exist within ICAP.com and ad.tullib.com forests. If it’s been removed, then it will need to be created using a root domain admin account in Sites and Services, and will have a single Replication link of high cost like “500”.



1. Create a new Subnet object stating the IP address of the Domain controller with a /32 at the end. This will make an entry of the single specific IP address. Select the “De-Advertised” site, and click OK.



1. Right click, and select properties of the newly created Subnet object and type the name of the Server in the Description Field.



1. Right click the DC and choose Move and select the Deadvertised site.
2. Reboot the domain controller.

## 1.4 Discover and Remove Statically assigned DNS dependencies

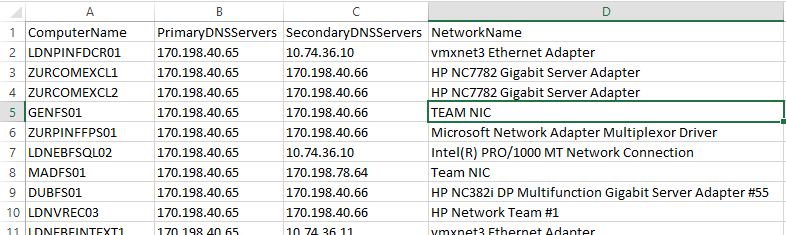
DNS is a fundamental protocol in modern infrastructure, and in its role as a DNS server, the domain controller being decommissioned might have statically assigned or hard-coded references on other network devices within the infrastructure. Efforts should be made to proactively identify these before the domain controller de-advertisement and monitoring begin.

### Windows Domain Joined DNS dependency Check

All domain joined Windows operating systems can be queried for the DNS servers statically assigned to them using the below script and analyzing the output. This script might potentially need to be run against multiple domains, but most likely running it within the domain that the DC resides would be sufficient. Save the below as Get-remotednsclientsetting.ps1.

|  |
| --- |
| [cmdletbinding()]  param (  [parameter(ValueFromPipeline=$true,ValueFromPipelineByPropertyName=$true)]  [string[]] $ComputerName = $env:computername  )  begin {}  process {  foreach($Computer in $ComputerName) {  Write-Verbose "Working on $Computer"  if(Test-Connection -ComputerName $Computer -Count 1 -ea 0) {    try {  $Networks = Get-WmiObject -Class Win32\_NetworkAdapterConfiguration `  -Filter IPEnabled=TRUE `  -ComputerName $Computer `  -ErrorAction Stop  } catch {  Write-Verbose "Failed to Query $Computer. Error details: $\_"  continue  }  foreach($Network in $Networks) {  $DNSServers = $Network.DNSServerSearchOrder  $NetworkName = $Network.Description  If(!$DNSServers) {  $PrimaryDNSServer = "Notset"  $SecondaryDNSServer = "Notset"  } elseif($DNSServers.count -eq 1) {  $PrimaryDNSServer = $DNSServers[0]  $SecondaryDNSServer = "Notset"  } else {  $PrimaryDNSServer = $DNSServers[0]  $SecondaryDNSServer = $DNSServers[1]  }  If($network.DHCPEnabled) {  $IsDHCPEnabled = $true  }    $OutputObj = New-Object -Type PSObject  $OutputObj | Add-Member -MemberType NoteProperty -Name ComputerName -Value $Computer.ToUpper()  $OutputObj | Add-Member -MemberType NoteProperty -Name PrimaryDNSServers -Value $PrimaryDNSServer  $OutputObj | Add-Member -MemberType NoteProperty -Name SecondaryDNSServers -Value $SecondaryDNSServer  $OutputObj | Add-Member -MemberType NoteProperty -Name IsDHCPEnabled -Value $IsDHCPEnabled  $OutputObj | Add-Member -MemberType NoteProperty -Name NetworkName -Value $NetworkName  $OutputObj    }  } else {  Write-Verbose "$Computer not reachable"  }  }  }  end {}  Run the following code in powershell window and call this script.  $computers = Get-ADComputer -Filter \* -Properties Name,OperatingSystem | Where-Object {($\_.OperatingSystem -like '\*Server\*')} | Select-Object Name  ./Get-remotednsclientsetting.ps1 -computername $computers.name | Export-CSV C:\temp\dnssettings.csv -NTI |

Example output below:



### Non-Windows, or Non-Domain Joined DNS dependency Check

This is difficult as there will be several devices within the infrastructure that we cannot remotely query for their statically assigned DNS server configuration or modify the statically assigned DNS server configuration. There are three options to identify these static assignments:

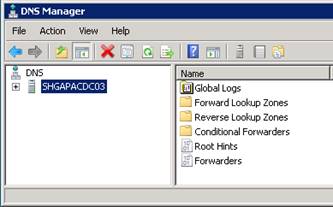
**1: Manual pro-active check**

Depending on the size of the site, manual checks could be completed on all network devices\appliances within the AD site subnet to ensure that static entries to de-advertised domain controllers are understood and changed if they exist. If this is a large site and, therefore, a large task that is not feasible, we will have to use one of the following methods.

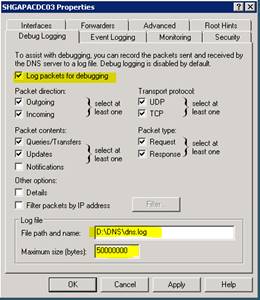
**2: DC Logging proactive check**

Parsing DNS server logs to trace active clients. The steps show how to enable debug logging of the DNS server service and parse the data to discover DNS queries.

1. Open the DNS Management console
2. Right-click the DNS server, click properties and go to the Debug Logging tab



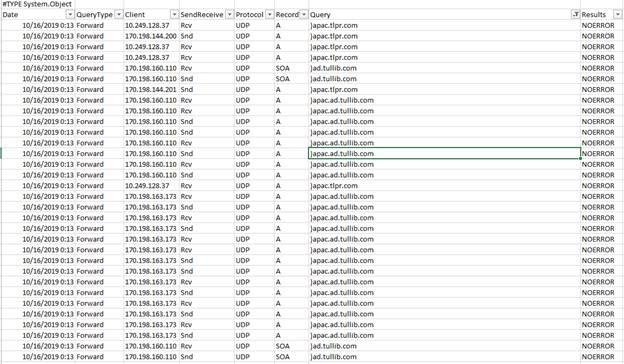
1. Type in the filename and path for the log file and change the maximum size (bytes), and click OK
2. Restart the DNS service



1. Parsing DNS Debug Log

|  |
| --- |
| function Get-DNSDebugLog  {      <#      .SYNOPSIS      This cmdlet parses a Windows DNS Debug log.      .DESCRIPTION      When a DNS log is converted with this cmdlet it will be turned into objects for further parsing.      .EXAMPLE      Get-DNSDebugLog -DNSLog ".\Something.log" | Format-Table      Outputs the contents of the dns debug file "Something.log" as a table.      .EXAMPLE      Get-DNSDebugLog -DNSLog ".\Something.log" | Export-Csv .\ProperlyFormatedLog.csv      Turns the debug file into a csv-file.      .PARAMETER DNSLog      Path to the DNS log or DNS log data. Allows pipelining from for example Get-ChildItem for files, and supports pipelining DNS log data.      #>      [CmdletBinding()]      param(        [Parameter(Mandatory=$false, ValueFromPipeline=$true, ValueFromPipelineByPropertyName=$true)]        [Alias('Fullname')]        [string] $DNSLog = 'StringMode')      BEGIN { }      PROCESS {          $TheReverseRegExString='\(\d\)in-addr\(\d\)arpa\(\d\)'          ReturnDNSLogLines -DNSLog $DNSLog | % {                  if ( $\_ -match '^\d\d|^\d/\d' -AND $\_ -notlike '\*EVENT\*' -AND $\_ -notlike '\* Note: \*') {                      $Date=$null                      $Time=$null                      $DateTime=$null                      $Protocol=$null                      $Client=$null                      $SendReceive=$null                      $QueryType=$null                      $RecordType=$null                      $Query=$null                      $Result=$null                      $Date=($\_ -split ' ')[0]                      # Check log time format and set properties                      if ($\_ -match ':\d\d AM|:\d\d  PM') {                          $Time=($\_ -split ' ')[1,2] -join ' '                          $Protocol=($\_ -split ' ')[7]                          $Client=($\_ -split ' ')[9]                          $SendReceive=($\_ -split ' ')[8]                          $RecordType=(($\_ -split ']')[1] -split ' ')[1]                          $Query=($\_.ToString().Substring(110)) -replace '\s' -replace '\(\d?\d\)','.' -replace '^\.' -replace "\.$"                          $Result=(((($\_ -split '\[')[1]).ToString().Substring(9)) -split ']')[0] -replace ' '                      }                      elseif ($\_ -match '^\d\d\d\d\d\d\d\d \d\d:') {                          $Date=$Date.Substring(0,4) + '-' + $Date.Substring(4,2) + '-' + $Date.Substring(6,2)                          $Time=($\_ -split ' ')[1] -join ' '                          $Protocol=($\_ -split ' ')[6]                          $Client=($\_ -split ' ')[8]                          $SendReceive=($\_ -split ' ')[7]                          $RecordType=(($\_ -split ']')[1] -split ' ')[1]                          $Query=($\_.ToString().Substring(110)) -replace '\s' -replace '\(\d?\d\)','.' -replace '^\.' -replace "\.$"                          $Result=(((($\_ -split '\[')[1]).ToString().Substring(9)) -split ']')[0] -replace ' '                      }                      else {                          $Time=($\_ -split ' ')[1]                          $Protocol=($\_ -split ' ')[6]                          $Client=($\_ -split ' ')[8]                          $SendReceive=($\_ -split ' ')[7]                          $RecordType=(($\_ -split ']')[1] -split ' ')[1]                          $Query=($\_.ToString().Substring(110)) -replace '\s' -replace '\(\d?\d\)','.' -replace '^\.' -replace "\.$"                          $Result=(((($\_ -split '\[')[1]).ToString().Substring(9)) -split ']')[0] -replace ' '                      }                     $DateTime=Get-Date("$Date $Time") -Format 'yyyy-MM-dd HH:mm:ss'                      if ($\_ -match $TheReverseRegExString) {                          $QueryType='Reverse'                      }                      else {                          $QueryType='Forward'                      }                      $returnObj = New-Object System.Object                      $returnObj | Add-Member -Type NoteProperty -Name Date -Value $DateTime                      $returnObj | Add-Member -Type NoteProperty -Name QueryType -Value $QueryType                      $returnObj | Add-Member -Type NoteProperty -Name Client -Value $Client                      $returnObj | Add-Member -Type NoteProperty -Name SendReceive -Value $SendReceive                      $returnObj | Add-Member -Type NoteProperty -Name Protocol -Value $Protocol                      $returnObj | Add-Member -Type NoteProperty -Name RecordType -Value $RecordType                      $returnObj | Add-Member -Type NoteProperty -Name Query -Value $Query                      $returnObj | Add-Member -Type NoteProperty -Name Results -Value $Result                      if ($returnObj.Query -ne $null) {                          Write-Output $returnObj                      }                  }              }      }      END { }  }  function ReturnDNSLogLines  {  param(  $DNSLog)  $PathCorrect=try { Test-Path $DNSLog -ErrorAction Stop } catch { $false }      if ($DNSLog -match '^\d\d|^\d/\d' -AND $DNSLog -notlike '\*EVENT\*' -AND $PathCorrect -ne $true) {          $DNSLog      }      elseif ($PathCorrect -eq $true) {          Get-Content $DNSLog | % { $\_ }      }  } |

1. Rename the DNSDebuglog.txt to .ps1
2. Run “Import-Module .\DNSDebug.ps1”
3. Run “Get-DNSDebugLog '.\dns.txt' | Export-Csv c:\temp\dns.csv”
4. Open the CSV file and filter for the domain where the domain controller resides.



**3: Manual reactive check**

A “controlled” reactive method of identifying statically assigned DNS resolver configuration dependencies on the domain controller or any other dependency. See section 4 of this document.

# 2 Pro-active Monitoring of De-Advertised Domain Controller – Perfmon Method

Following on from the de-advertising process and waiting a period (at least one working week but probably over eight days-DHCP lease time) for traffic to fall to a minimum base level, we now want to see what is still contacting the Domain Controller. Clients contacting the server now are probably doing so because they are hard coded with references to the domain controller by name or IP address.

The PERFMON method provides the following services, providing specific detailed information from the requests themselves:

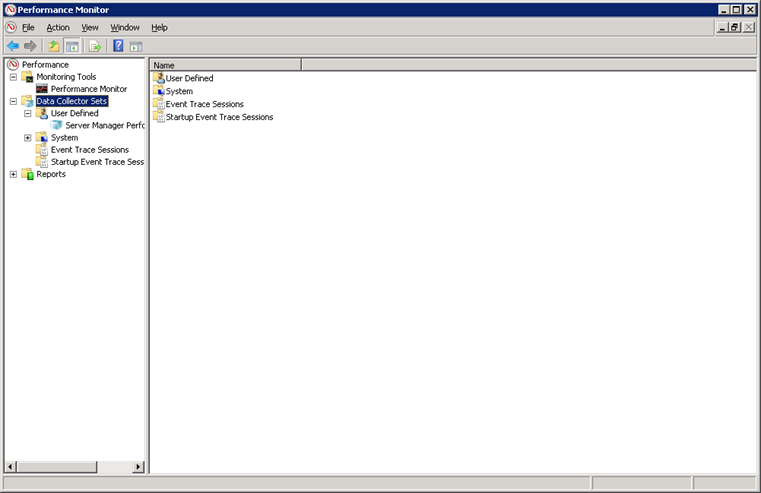
* LDAP queries
* Kerberos authentication requests
* NTLM authentication requests

Perform the following steps to capture this.

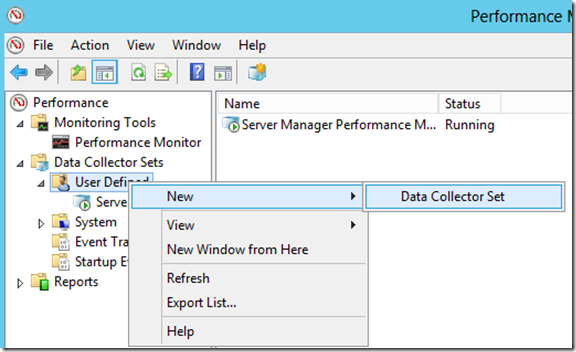
## 2.1 Configure Performance Monitor

1. Run Performance Monitor by going Start – Run – Perfmon.

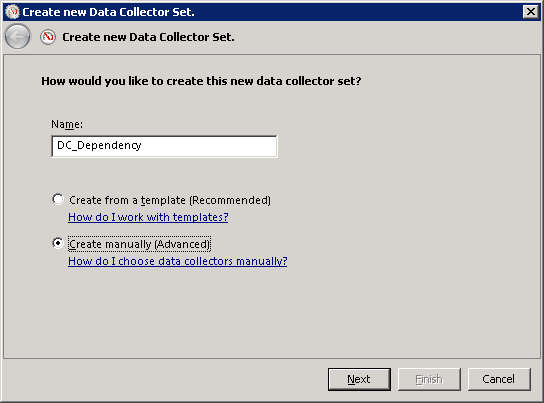
Then expand Data Collector Sets:



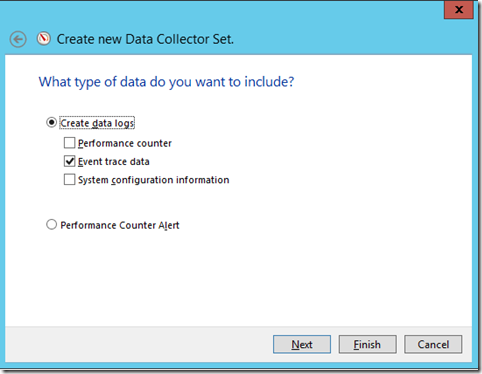
1. Right click User Defined – New – Data Collector Set

[](https://msdnshared.blob.core.windows.net/media/TNBlogsFS/prod.evol.blogs.technet.com/CommunityServer.Blogs.Components.WeblogFiles/00/00/00/91/74/metablogapi/image_41EC7435.png)

1. Type DC\_Dependency as the name of the DCS, select Create Manually (Advanced) and click next:



1. Select Event Trace Data:

[](https://msdnshared.blob.core.windows.net/media/TNBlogsFS/prod.evol.blogs.technet.com/CommunityServer.Blogs.Components.WeblogFiles/00/00/00/91/74/metablogapi/image_6EBC6A14.png)

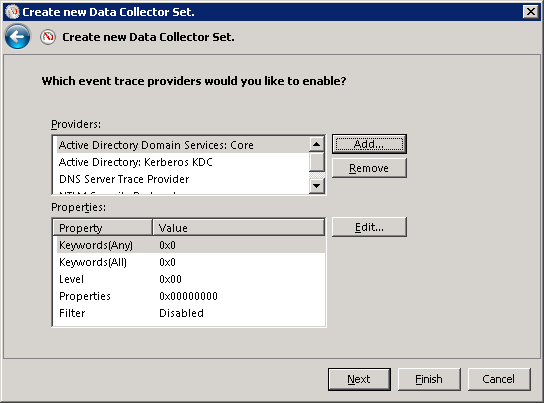
1. On the Event Providers click add and select the following providers:

Active directory Domain services: Core

Active Directory: Kerberos KDC

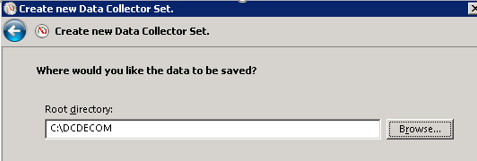
DNS Server Trace Provider

NTLM Security Protocol



1. Click next, select the path to save this file, and click finish.

e.g. C:\DCDECOM or D:\DCDECOM, on whichever has ~6GB or more

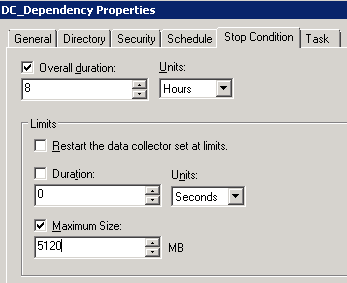


1. On the Perfmon console right, click your newly created DCS “DC\_Dependency” and select properties:
2. Select the “Stop Condition” tab and set

Overall duration (Check box and set to) = 8 Hours

Maximum Size (Check box and set to) = 5120 MB

Click Ok



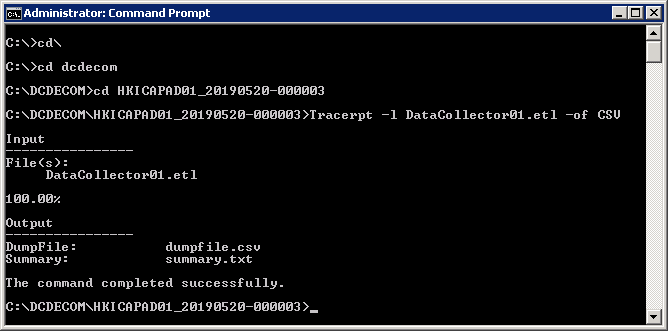
1. Wait at least one full working day for results to be captured. Longer than this may create very large, and therefore difficult to analyze, results.

## 2.2 Collect and Convert Results

1. After the data is collected you’ll have a file called datacollector01.etl. This file is not in a readable format. We need to convert the file into a readable .CSV format
2. Open an elevated command prompt on the Domain controller
3. Navigate to the Data Collection subdirectory and run the following command:

Tracerpt -l DataCollector01.etl -of CSV

1. Do not run this command on other machine as the providers might not be available and the events decoding will be incomplete.



1. When the process is done, you’ll have 2 files:

Summary.csv

You can use this file to validate that all the providers were found, to do that just check that all rows on the event name column are populated, if you have one or more empty rows, the system you used does not have the correct providers (which shouldn’t happen if you run this on the DC since it has all of them)

Dumpfile.csv:

Containing the output data from the data collection capture in a readable format, however the file will be very large and contain numerous and repeated entries for subscribing clients.

1. The contents of the large Dumpfile.csv can be converted to a readable format by using an Excel macro with pre-built pivots. Copy the dumpfile to a machine with Excel installed, copy “Import-DC\_Info.xlsm” to the same directory where the “dumpfile.csv” is located, open it and enable macros, then click Import file.



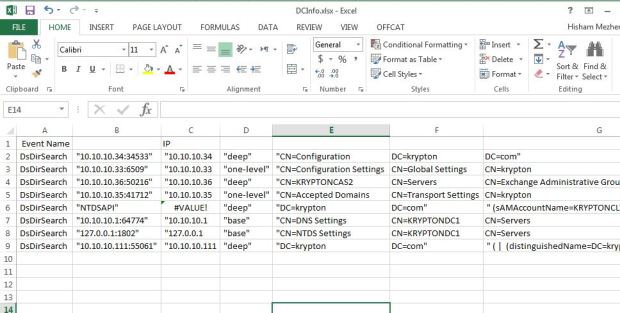
## 2.3 Analyse Results

After this process is done you’ll have an Excel file called DCInfo.xlsx with some tabs and pivot tables created. Below describes each tab and what information it provides.



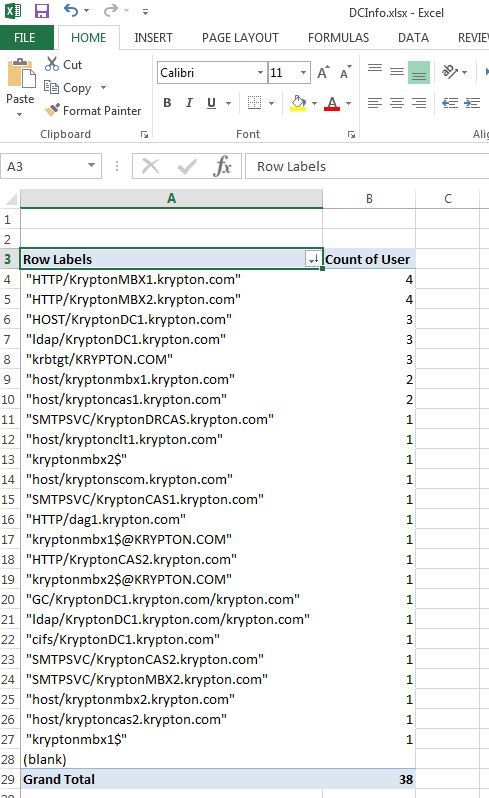
**LDAP**

It contains a list of all the LDAP queries performed against our DC with a list of IP:Port combination and also the query that was executed, with this you can see who is requesting what info and from what IP this query was originated.



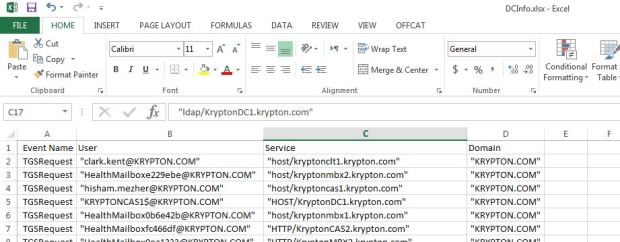
**Kerberos-Pivot**

This is a Pivot table populated from the Kerberos tab which is sorted by the total number of hits to a particular service, this table is helpful to have a quick glance of what service is still using Kerberos authentication.



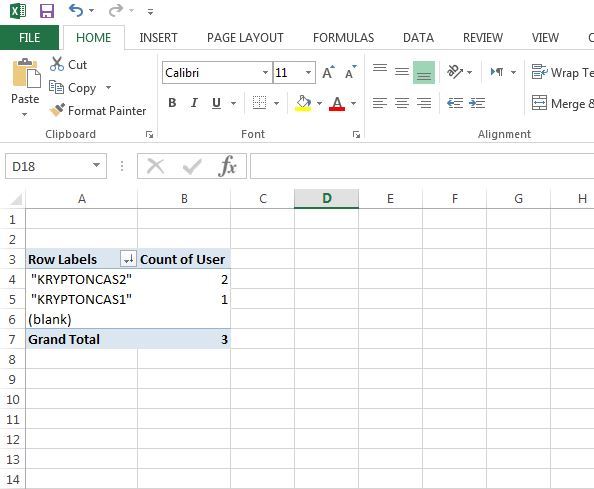
**Kerberos**

This table is a consolidated list of all the TGS requests made, we’ll use this tab if you want to see more detail from the pivot table, more specifically the User column will show you the actual user/computer/service account that is requesting a service while the service column indicates what service is being requested.



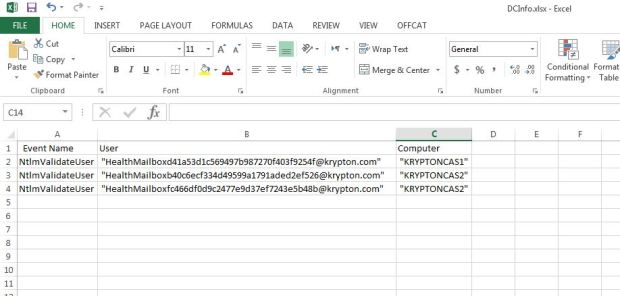
**NTLM-Pivot**

This table is very similar to the Kerberos-Pivot, it will give us a list of the total number of NTLMValidateUser requests being performed from clients to services.



**NTLM**

A full list of every NTLMValidateUser requests, similar to the Kerberos tab.



Investigate with the Application Server owners/support to remove the hardcoded references ahead of the next stage.

References:

<https://techcommunity.microsoft.com/t5/Core-Infrastructure-and-Security/Domain-and-DC-Migrations-How-To-Monitor-LDAP-Kerberos-and-NTLM/ba-p/256796>

<https://hishammezher.wordpress.com/2017/05/21/how-to-identify-and-monitor-ldap-kerberos-and-ntlm-connections-to-a-domain-controller/>

# 3 Pro-active Monitoring of De-Advertised Domain Controller – Connection Logger Method

This monitor method looks only at network connections (so source IP and port), and so can cover a broader range of services with potentially fewer details. The services covered are as follows:

* DNS lookup (UDP 53)
* LDAP queries (TCP\UDP 389)
* Secure LDAP queries (TCP 636)
* Kerberos authentication requests (TCP\UDP 88)
* NTP requests (UDP 123)
* Global Catalog requests (TCP 3268)
* Secure Global Catalog requests (TCP 3269)
* SMB requests (TCP 445)
* SYSVOL replication (TCP 139)

Perform the following steps to capture activity using this method.

## 3.1 Configure Connection Logger

Connection Logger is a console-based x86 utility written by MCS that runs on all versions of Windows from XP to Server 2012 R2. It should be used alongside the Perfmon method or as an alternative if the domain controller is in a branch office.

Benefits

* Minimal executable footprint on the target server
* No need to install drivers
* Minimal logging footprint on the target server
* Logs only single-instance of source IP
* DB size configurable as “space-to-leave” on install volume
* Specific targeted logging
* Listen only on specified ports

The application is made up of the following files:

* ConnectionLogger.exe
* ConnectionLog.ini
* ConnectionLogExporter.exe

Copy the files from [\\emearepository\install$\Applications\ConnectionLogger](file:///\\emearepository\install$\Applications\ConnectionLogger) to the domain DCs that are being demoted, recommended to use the folder “D:\ConnectionLogger” however the app has xontrols around the amount of space it can consume so it’s safe to deploy to C:\ drive if no other disk is available.

The app can be run interactively within a console or RDP session, or if it needs to be run for a long period of time out of hours and these sessions would be logged off after idle timeout, can be set to run as SYSTEM, run as scheduled task on startup

schtasks /create /sc ONSTART /tn ConnectionLogger /tr C:\ConnectionLogger\ConnectionLogger.exe /ru System

Configure the ini file with the following content.

ListenOnIPAddress=10.10.12.11 *(IP address of DC)*

MonitorInterval=5000 ()

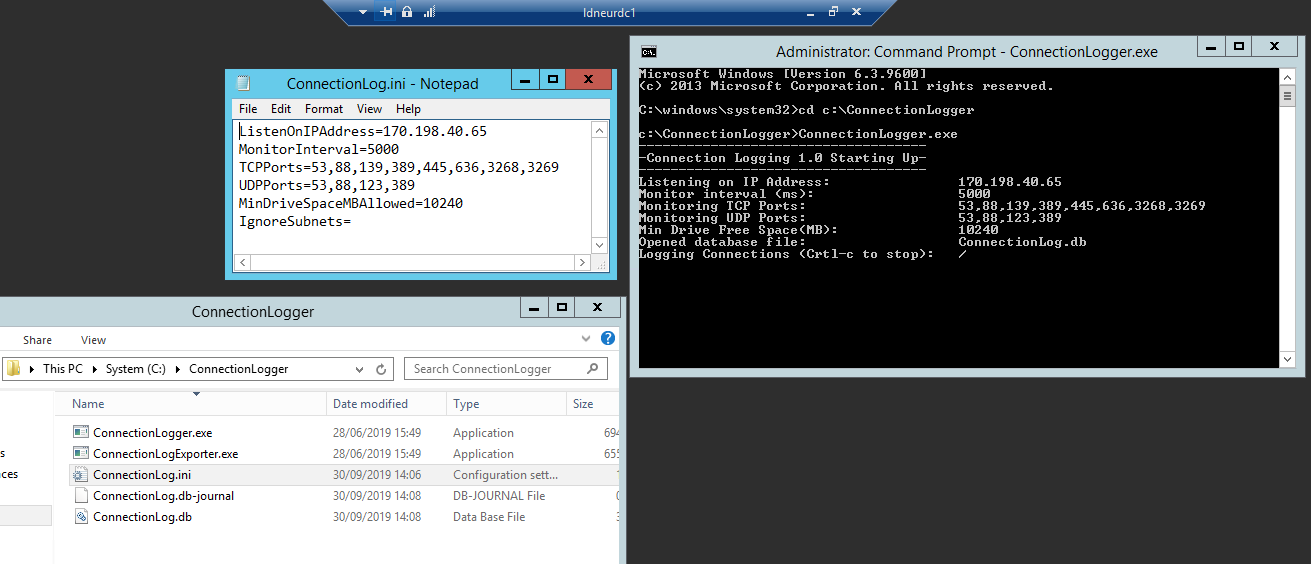
TCPPorts=53,88,139,389,445,636,3268,3269

UDPPorts=53,88,123,389

MinDriveSpaceMBAllowed=10240

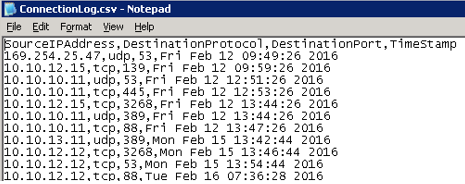
IgnoreSubnets=

See example below of configuration running interactively within an RDP session.



## 3.2 Collect Connection Logger results

Once the ConnectionLogger is running it will continue to log unique sourceIP\port connection combinations into the ConnectionLog.db file. These can be exported into a CSV file using ConnectionLogExporter.exe anytime. Running this will export everything logged in the .db file into CSV. Example below:



## 3.3 Analyse Connection Logger Results

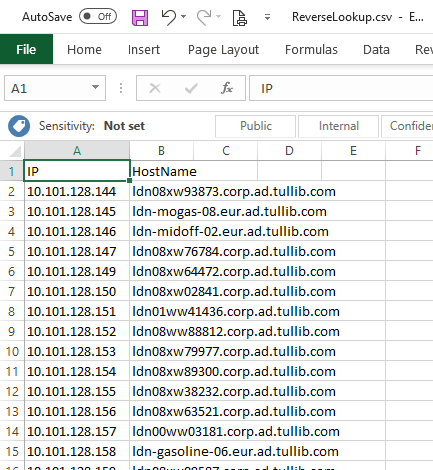
The csv is already pretty readable, however, there are several steps that can assist further by converting IP and port numbers to names. To analyze the content of the CSV we need to perform several steps as follows:

* Generate an IP to name list
* Generate a Port number to name list
* Add columns to the original ConnectionLog.csv and perform vlookup to pull names into the output.

### Generate IP to name list.

From the csv file, take the IP address column and place it in its own csv file called IP\_List.csv. Then run the following PowerShell code to perform reverse IP lookup. The ReverseLookup.csv file is the result.

|  |
| --- |
| $IPADDRS = Get-Content "C:\temp\ConnectionLogs\IP\_List.txt"  $result = @()  ForEach ($IPADDR in $IPADDRS)  {$result += [System.Net.DNS]::GetHostbyAddress($IPADDR) | Add-Member -Name IP -Value $IPADDR -MemberType NoteProperty -PassThru | Select-Object IP, HostName}  $result | Sort-Object -property IP | export-csv "C:\temp\ConnectionLogs\ReverseLookup.csv" -NoTypeInformation |



### Generate Port number to name list.

Copy the below into a new Excel sheet and save it as PortNames.xlsx.

|  |  |  |  |
| --- | --- | --- | --- |
| DestinationProtocol | DestinationPort | FullProtocolPort | ServiceName |
| udp | 53 | udp 53 | DNS resolver requests |
| tcp | 53 | tcp 53 | DNS Server zone transfers |
| tcp | 389 | tcp 389 | ldap |
| udp | 389 | udp 389 | ldap |
| tcp | 88 | tcp 88 | Kerberos auth |
| udp | 88 | udp 88 | Kerberos auth |
| tcp | 636 | tcp 636 | Secure ldap |
| udp | 123 | udp 123 | NTP requests |
| tcp | 3268 | tcp 3268 | GC clients |
| tcp | 3269 | tcp 3269 | GCs clients |
| tcp | 445 | tcp 445 | smb |
| tcp | 139 | tcp 139 | SYSVOL replication |

### Add columns containing Vlookup to ConnectionLog.csv

Pull the data generated in the previous steps into the ConnectionLog.csv file using the below vlookups.

Add a column to the sheet and then to add source hostname use the below vlookup:

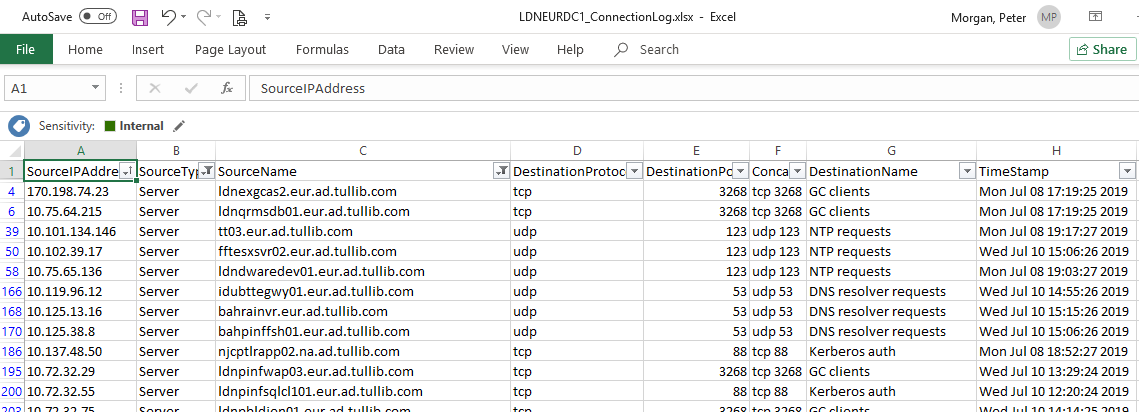
|  |
| --- |
| =VLOOKUP(A:A,'C:\temp\ConnectionLogs\[ReverseLookup.csv]ReverseLookup'!$A$1:$B$5000,2,FALSE) |

Ensure that the second number (currently at 5000) is at least as high as the number of rows in the sheet or you can miss some off.

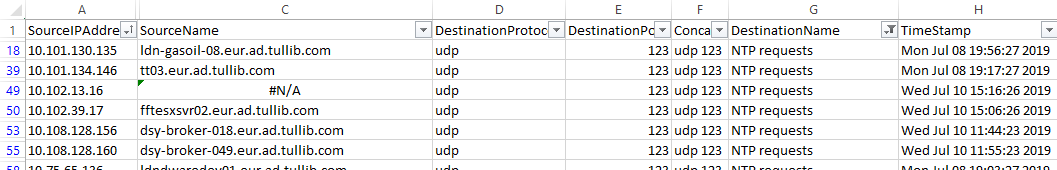
To add destination port name use the following:

|  |
| --- |
| =VLOOKUP(F:F,'C:\temp\ConnectionLogs\[PortNames.xlsx]Sheet1'!$C$2:$D$13,2,FALSE) |

Once completed and saved in Excel format your sheet will look like this:



You will then be able to filter by “SourceName” or by “DestinationName” (service being used) to identify where hard coded dependencies exist. For example to see where NTP requests are sent to the DC we filter by destination name “NTP requests”. This then shows all devices on the network still querying the domain controller for time.



Any #N\A entries mean there was no reverse lookup record for the IP address on the DNS server you queried. The records may not exist anywhere, or they may exist within a different domain so you could run this in several domains in order to get the most successful reverse lookups.

**Drawbacks**

As the ConnectionLogger app only logs a single instance of per IP per Port connections incoming to a domain controller, you do not get weighted information, or a count, of requests from any source. All connections are logged as equal. The reality is one source may have sent 10,000s of requests for a particular service where others may only have sent 1, but they will look the same in the output with a single line each. This is done in order for the output to be a manageable size.

# 4 Reactive Monitoring of De-Advertised Domain Controller

In some cases, it may not be possible to proactively identify all services required by a hard-coded domain controller's IP or name. Reasons for this might include:

* Too many connections are being made for a specific service to break it down and trace back the important connections.
* The requested services are not monitored by any method within this document.

Once efforts have been made to proactively address any dependency identified, and if there remains doubt that there are still dependencies that have not been identified, then there is the option to perform a reactive monitoring method.

Shut the DC down and see what breaks. If anything is revealed, the domain controller can be powered back up, and the now-revealed dependency can be investigated and rectified. The domain controller can then be powered down again. The domain controller should remain powered off for at least a week with no new issues reported before powering back ON and then proceeding with the demotion steps.

# 5 Demote Domain Controller

## 5.1 Pre Demotion checks

Once satisfied that any clients that are directly dependent on the domain controller being demoted have been identified and resolved or rationalized, proceed with the below.

### Global Catalog Check

A domain controller configured as a Global Catalog server (GC) stores the objects from all domains in the forest. For each object that is not directly in the domain for which the global catalog server belongs, a limited set of attributes is stored in the partial replica of the domain. Therefore, a global catalog server stores its own full, writable domain replica (all objects and all attributes) and the partial, read-only replica of other domains in the forest. Note: Don’t remove the last global catalog server from a domain. This will cause serious problems.

If the domain controller is a GC, ensure at least one other GC is configured within the domain before proceeding.

### FSMO Role Holder Check

If you followed the “De-advertise” process earlier in the document, you will already have moved the FSMO role to another DC. If the process was skipped, you need to do this now.

Operations master roles servers are designated to perform specific tasks to ensure the consistency of the entries in the AD database. AD DS has 5 Operation master roles. Note: Don’t demote the domain controller holding a domain-wide or forest-wide operations master role. Make sure to transfer the role to the other server before removing it.

The following operations masters perform operations that must occur on only one domain controller in the forest:

• Schema master

• Domain naming master

The following operations masters perform operations that must occur on only one domain controller in a domain:

• Primary Domain Controller (PDC) emulator

• Infrastructure master

• Relative ID (RID) master

### DNS Server Forwarding Check

The demotion process will check all other domain controllers in the forest to see if it has been specified as a DNS server forwarder. However should you wish, you can complete this check manually using the following powershell module, usage can be found in the link:

<https://gallery.technet.microsoft.com/Additional-PowerShell-09879e8d>

### Conditional Forward Zone – Cross Forest Check

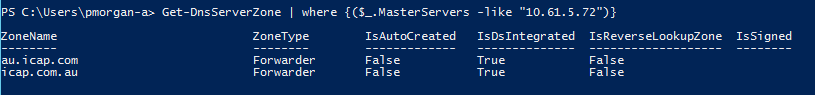
Check if the DC is being used on another DNS server as Master DNS Server in a conditional forwarder for cross-forest DNS lookup for a zone it hosts. Note: If the server is the last active server in a conditional forward zone then lookups from the machine in one domain to other will potentially break any external trust between the domains. Conditional forward lookup zones are stored in AD and therefore will be the same on all domain controllers in the forest.

If it’s an ICAP.com DC then the above command should be run on one DC in each domain in ad.tullib.com, and the reverse in the opposite case. Run the following on a DC in the root domain of the other forest.

|  |
| --- |
| Get-DnsServerZone | where {$\_.MasterServers -like "\*ipaddressofDCbeingremoved"} |

In most cases you’ll get 0 results, but if a result is found then another DC should be placed into the conditional forwarder to replace the outgoing one (to a max of 3 servers per forwarder zone).

As an example, let’s say we were demoting AUICAPAD08.au.icap.com (10.61.5.72) for some reason. Running the above command on a CORP DC shows it’s specified in the following conditional forwarder zone.



### DNS Server For DHCP Clients Check

If you followed the “De-advertise” process earlier in the document you will already have checked that this server is not being referenced as a DNS server within a DHCP scope. However, if the process was skipped you need to do this now. Please refer to section 1.1 for DHCP, and 1.4 for static assignments.

## 5.2 Demotion Processes

Depending on the OS version that the domain controller is installed upon will dictate which of the next sections to follow.

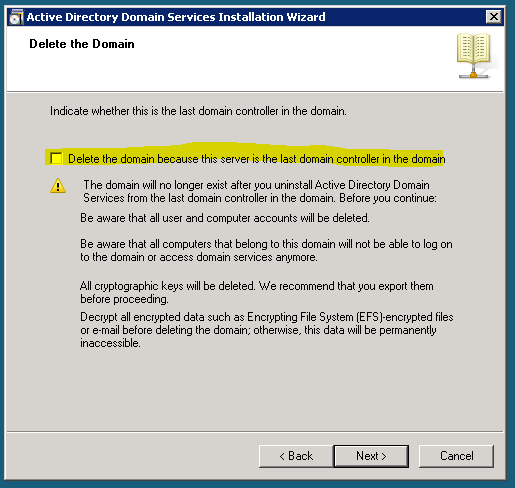
### 5.2.1 Windows 2008 R2 Steps

#### Demoting the DC from AD DS

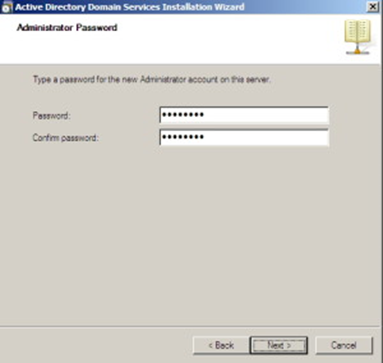
1. Start the Active Directory Domain services Installation Wizard by running “**dcpromo**” from the, **Start** Search box, and pressing Enter. Click next on the welcome screen.
2. If the Domain controller is a global catalog server, a message appears as warning to make sure that the other global catalog servers are available. Click OK on the prompt.



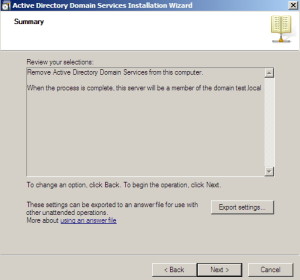
1. On the Delete the Domain, Click Next without making a selection, not unless the domain controller you are decommissioning is the last DC in the domain.

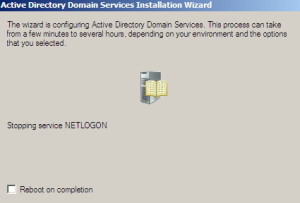


1. Enter the password for the Domain “Administrator” account (refer to Keepass) and click Next



1. Click **Next** and wait until

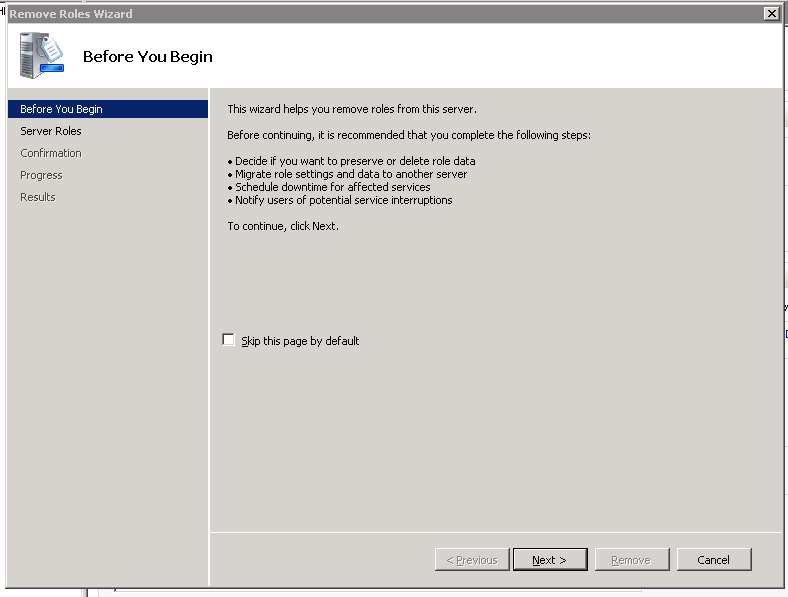




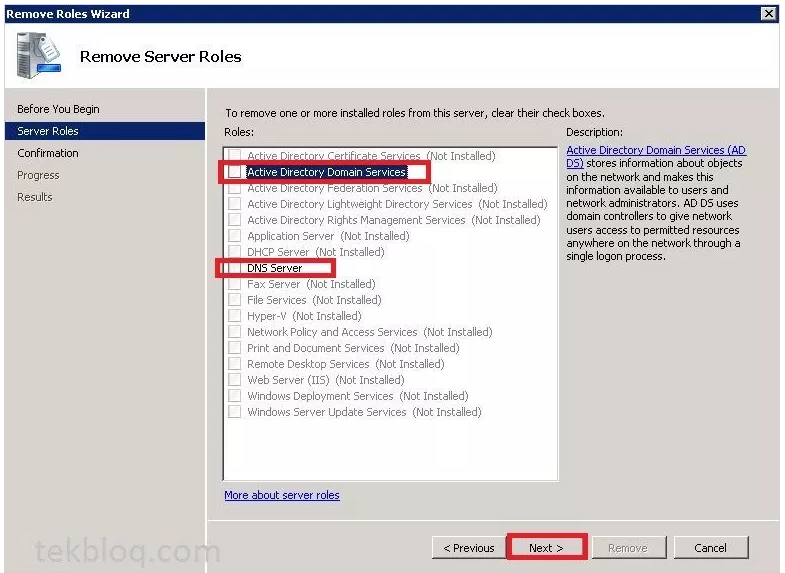
1. Click **Finish,** then the server reboots

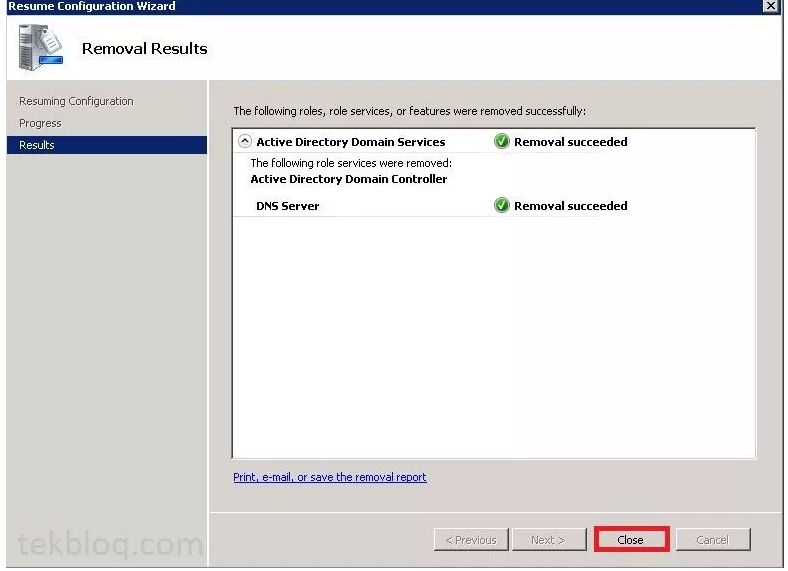
#### Remove DNS and AD DS roles

1. After reboot, Login to the server using local admin account. Open Server Manager. Click Start, point to Administrative Tools, and then click Server Manager and In Roles, click Remove Roles.



1. De-select “Active Directory Domain Services” and “DNS Sever” roles, and proceed to the end of the wizard.





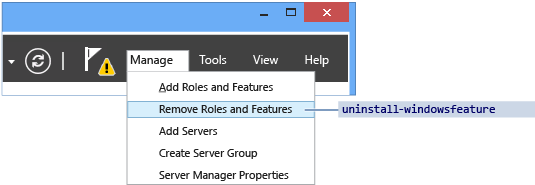
1. Reboot the server, and complete verification post check steps in section 3.2.3.

### 5.2.2 Windows 2012 or later Steps

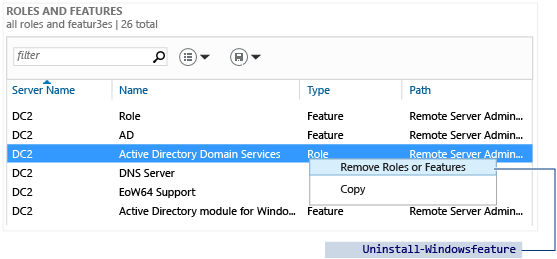
#### Demoting the DC from AD DS

Server Manager offers two interfaces to removing the Active Directory Domain Services role:

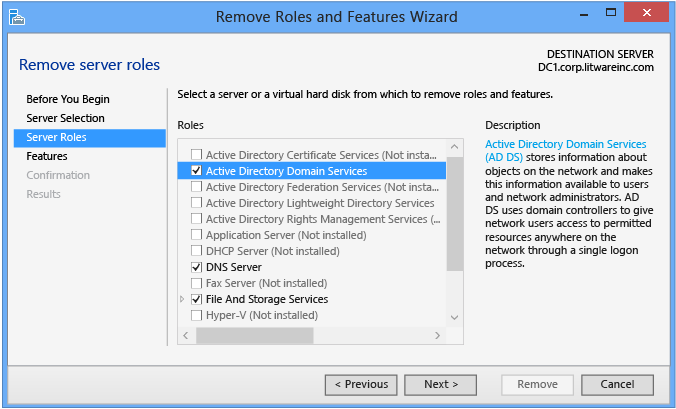
The **Manage** menu on the main dashboard, using **Remove Roles and Features**



Click **AD DS** or **All Servers** on the navigation pane. Scroll down to the **Roles and Features** section. Right-click **Active Directory Domain Services** in the **Roles and Features** list and click **Remove Role or Feature**. This interface skips the **Server Selection** page.



#### Server Roles and Features

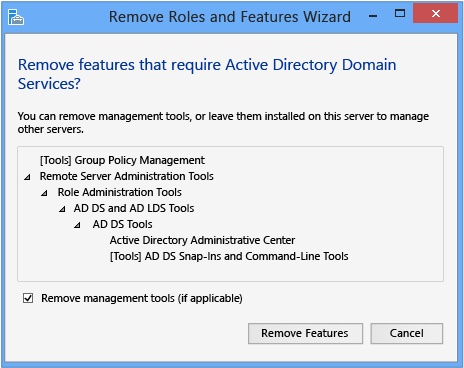


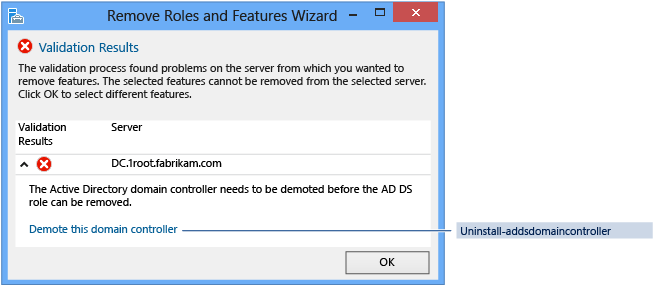
Clear the **Active Directory Domain Services** check box to demote a domain controller; if the server is currently a domain controller, this does not remove the AD DS role and instead switches to a **Validation Results** dialog with the offer to demote. Otherwise, it removes the binaries like any other role feature.

* Do not remove any other AD DS-related roles or features - such as DNS, GPMC, or the RSAT tools - if you intend to promote the domain controller again immediately. Removing additional roles and feature increases the time to re-promote, as Server Manager reinstalls these features when you reinstall the role.
* Remove unneeded AD DS roles and features at your own discretion if you intend to demote the domain controller permanently. This requires clearing the check boxes for those roles and features.

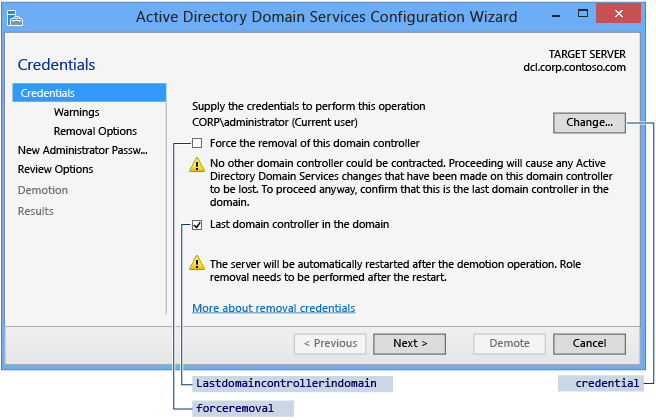
The full list of AD DS-related roles and features include:

* + Active Directory Module for Windows PowerShell feature
  + AD DS and AD LDS Tools feature
  + Active Directory Administrative Center feature
  + AD DS Snap-ins and Command-line Tools feature
  + DNS Server
  + Group Policy Management Console





#### Credentials

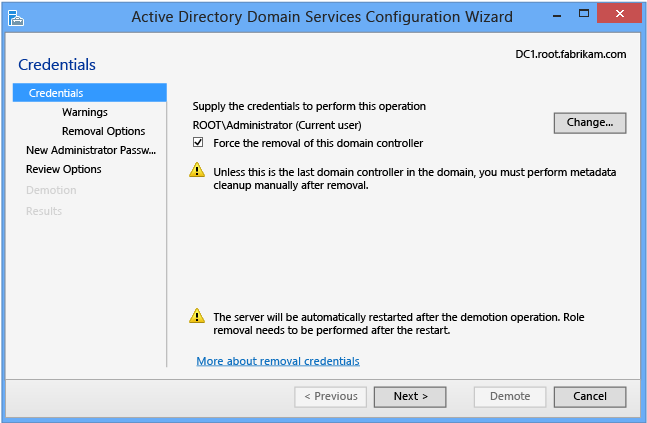


You configure demotion options on the **Credentials** page. Provide the credentials necessary to perform the demotion from the following list:

* Demoting an additional domain controller requires Domain Admin credentials. Selecting **Force the removal of this domain controller** demotes the domain controller without removing the domain controller object's metadata from Active Directory.

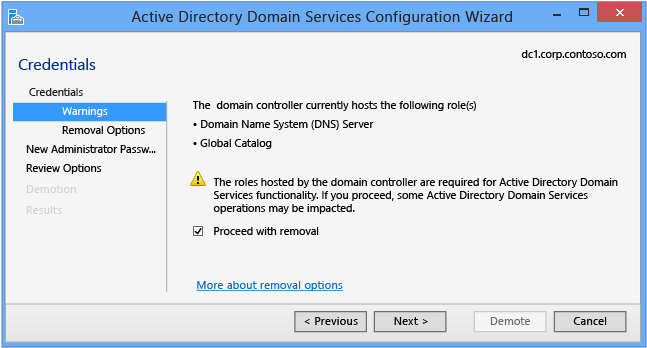
Warning: Do not select this option unless the domain controller cannot contact other domain controllers and there is no reasonable way to resolve that network issue. Forced demotion leaves orphaned metadata in Active Directory on the remaining domain controllers in the forest. In addition, all un-replicated changes on that domain controller, such as passwords or new user accounts, are lost forever. Orphaned metadata is the root cause in a significant percentage of Microsoft Customer Support cases for AD DS, Exchange, SQL, and other software.

If you forcibly demote a domain controller, you must manually perform metadata cleanup immediately. For steps, review [Clean Up Server Metadata](https://technet.microsoft.com/library/cc816907(WS.10).aspx).



* Demoting the last domain controller in a domain requires Enterprise Admins group membership, as this removes the domain itself (if the last domain in the forest, this removes the forest). Server Manager informs you if the current domain controller is the last domain controller in the domain. Select the **Last domain controller in the domain** check box to confirm the domain controller is the last domain controller in the domain.

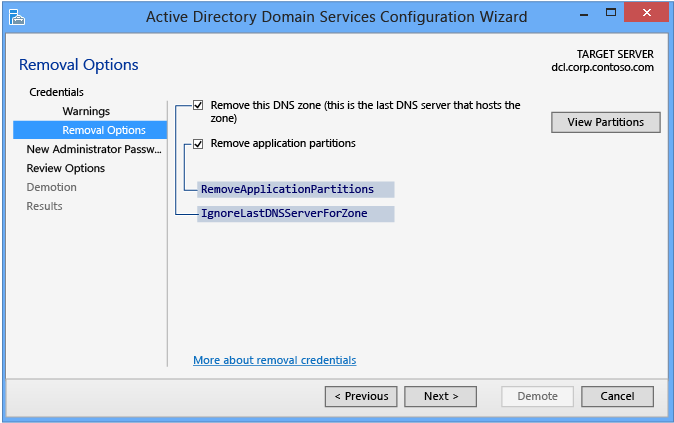
#### Warnings



The **Warnings** page alerts you to the possible consequences of removing this domain controller. To continue, you must select **Proceed with removal**.

 Warning: If you previously selected **Force the removal of this domain controller** on the **Credentials** page, then the **Warnings** page shows all Flexible Single Master Operations roles hosted by this domain controller. You mustseize the roles from another domain controller immediately after demoting this server. For more information on seizing FSMO roles, see [Seize the Operations Master Role](https://technet.microsoft.com/library/cc816779(WS.10).aspx).

#### Removal Options

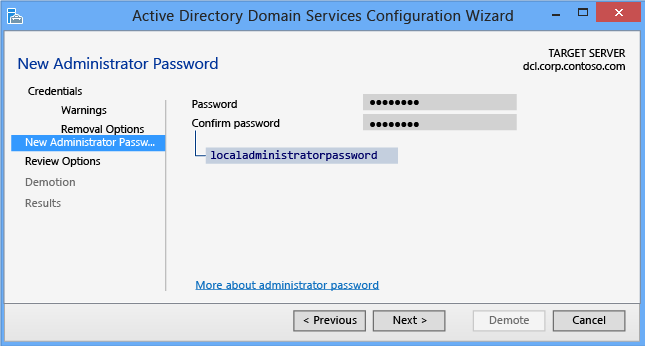


The **Removal Options** page appears depending on previously selecting **Last domain controller in the domain** on the **Credentials** page. This page enables you to configure additional removal options. Select **Ignore last DNS server for zone**, **Remove application partitions**, and **Remove DNS Delegation** to enable the **Next** button.

The options only appear if applicable to this domain controller. For instance, if there is no DNS delegation for this server then that checkbox will not display.

Click **Change** to specify alternate DNS administrative credentials. Click **View Partitions** to view additional partitions the wizard removes during the demotion. By default, the only additional partitions are Domain DNS and Forest DNS Zones. All other partitions are non-Windows partitions.

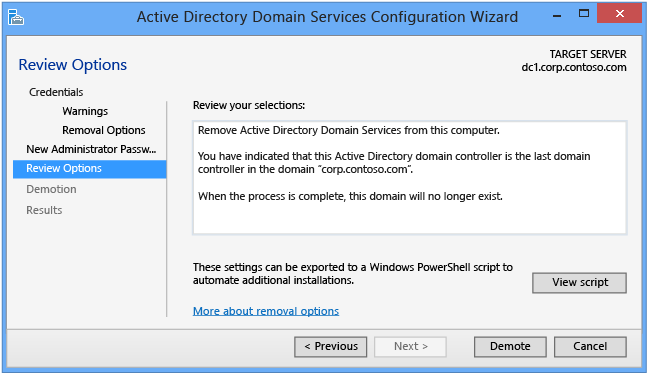
#### New Administrator Password



The **New Administrator Password** page requires you to provide a password for the built-in local computer's Administrator account, once the demotion completes and the computer becomes a domain member server or workgroup computer.

Warning: As the previous two options do not confirm the password, use extreme caution: the password is not visible.

#### Confirmation

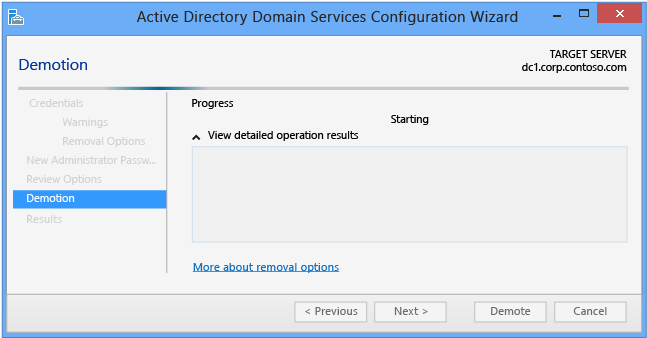


The **Confirmation** page shows the planned demotion; the page does not list demotion configuration options. This is the last page the wizard shows before the demotion begins. The View Script button creates a Windows PowerShell demotion script.

Click **Demote** to run the following AD DS Deployment cmdlet:

The prompt to restart is your last opportunity to cancel this operation when using ADDSDeployment Windows PowerShell. To override that prompt, use the **-force** or **confirm:$false** arguments.

#### Demotion



When the **Demotion** page displays, the domain controller configuration begins and cannot be halted or canceled. Detailed operations are displayed on this page and written to logs:

* %systemroot%\debug\dcpromo.log
* %systemroot%\debug\dcpromoui.log

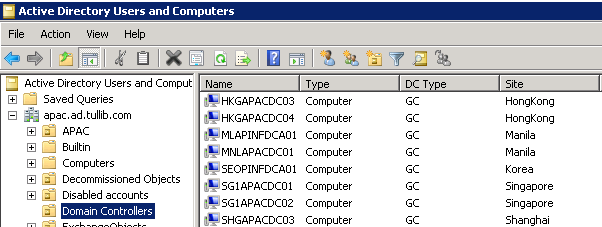
The **Results** page shows the success or failure of the promotion and any important administrative information. The domain controller will automatically reboot after 10 seconds.

#### Remove the DNS and AD DS roles

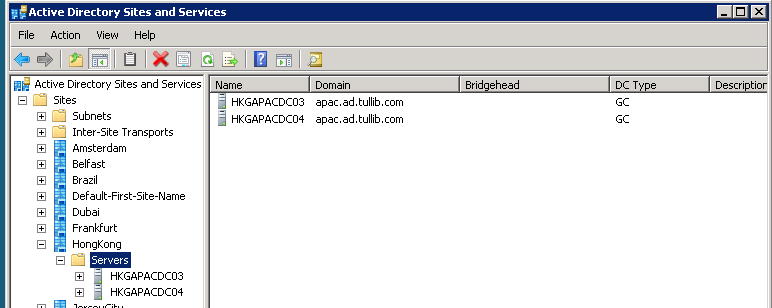
TBC

### 5.2.3 Verification Post Check – All Versions

1. Login to an existing Domain controller in the domain and check that the server you decommission has been removed from the “Domain Controller” OU. Example below:



1. Open the Active Directory Sites and Services. Navigate the Site name of the server you decommission. Right Click the server and delete. Example below:



# 6 TroubleShooting Domain Controller Demotion

There are occasions when the demotion process cannot be completed or does not proceed to completion correctly for some undiscernible reason. In these instances, a small effect can be made to understand and rectify the issue; however, sometimes it’s faster to follow the below processes to force remote the domain controller and perform a manual cleanup.

## 6.1 Known Issues

Use DcPromo /forceremoval and then perform meta-data cleanup steps.

## 6.2 Clean up metadata using GUI tools

Metadata cleanup is a required procedure after a forced removal of Active Directory Domain Services (AD DS). You perform metadata cleanup on a domain controller in the domain of the domain controller that you forcibly removed. Metadata cleanup removes data from AD DS that identifies a domain controller to the replication system. Metadata cleanup also removes File Replication Service (FRS) and Distributed File System (DFS) Replication connections and attempts to transfer or seize any operations master (also known as flexible single master operations or FSMO) roles that the retired domain controller holds.

Note: If you receive an “Access is denied” error when you use any of these methods to perform metadata cleanup, make sure that the computer object and the NTDS Settings object for the domain controller are not protected against accidental deletion. To verify this right-click the computer object or the NTDS Settings object, click Properties, click Object, and clear the Protect object from accidental deletion check box. In Active Directory Users and Computers, the Object tab of an object appears if you click View and then click Advanced Features.

When you use Remote Server Administration Tools (RSAT) or the Active Directory Users and Computers console (Dsa.msc) that is included with Windows Server to delete a domain controller computer account from the Domain Controllers organizational unit (OU), the cleanup of server metadata is performed automatically. Before Windows Server 2008, you had to perform a separate metadata cleanup procedure.

You can also use the Active Directory Sites and Services console (Dssite.msc) to delete a domain controller’s computer account, which also completes metadata cleanup automatically. However, Active Directory Sites and Services removes the metadata automatically only when you first delete the NTDS Settings object below the computer account in Dssite.msc.

As long as you are using the Windows Server 2008 or newer RSAT versions of Dsa.msc or Dssite.msc, you can clean up metadata automatically for domain controllers running earlier versions of Windows operating systems.

Membership in Domain Admins, or equivalent, is the minimum required to complete these procedures.

### Using Active Directory Users and Computers

Reference: <https://docs.microsoft.com/en-us/windows-server/identity/ad-ds/deploy/ad-ds-metadata-cleanup#clean-up-server-metadata-using-activedirectory-users-and-computers>

1. Open Active Directory Users and Computers.
2. If you have identified replication partners in preparation for this procedure and if you are not connected to a replication partner of the removed domain controller whose metadata you are cleaning up, right-click Active Directory Users and Computers node, and then click Change Domain Controller. Click the domain controller name from which you want to remove the metadata, and then click OK.
3. Expand the domain controller's domain that was forcibly removed, and then click Domain Controllers.
4. In the details pane, right-click the computer object of the domain controller whose metadata you want to clean up, and then click Delete.
5. In the Active Directory Domain Services dialog box, confirm the name of the domain controller you wish to delete is shown, and click Yes to confirm the computer object deletion.
6. In the Deleting Domain Controller dialog box, select This Domain Controller is permanently offline and can no longer be demoted using the Active Directory Domain Services Installation Wizard (DCPROMO), and then click Delete.
7. If the domain controller is a global catalog server, in the Delete Domain Controller dialog box, click Yes to continue with the deletion.
8. If the domain controller currently holds one or more operations master roles, click OK to move the role or roles to the domain controller that is shown. You cannot change this domain controller. If you want to move the role to a different domain controller, you must move the role after you complete the server metadata cleanup procedure.

### Using Active Directory Sites and Services

Reference: <https://docs.microsoft.com/en-us/windows-server/identity/ad-ds/deploy/ad-ds-metadata-cleanup#clean-up-server-metadata-using-activedirectory-sites-and-services>

1. Open Active Directory Sites and Services.
2. If you have identified replication partners in preparation for this procedure and are not connected to a replication partner of the removed domain controller whose metadata you are cleaning up, right-click Active Directory Sites and Services, then click Change Domain Controller. Click the domain controller name from which you want to remove the metadata, and then click OK.
3. Expand the site of the domain controller that was forcibly removed, expand Servers, expand the domain controller's name, right-click the NTDS Settings object, and then click Delete.
4. In the Active Directory Sites and Services dialog box, click Yes to confirm the deletion of the NTDS Settings.
5. In the Deleting Domain Controller dialog box, select This Domain Controller is permanently offline and can no longer be demoted using the Active Directory Domain Services Installation Wizard (DCPROMO), and then click Delete.
6. If the domain controller is a global catalog server, click Yes in the Delete Domain Controller dialog box to continue with the deletion.
7. If the domain controller currently holds one or more operations master roles, click OK to move the role or roles to the shown domain controller.
8. Right-click the domain controller that was forcibly removed, and then click Delete.
9. Click Yes to confirm the domain controller deletion in the Active Directory Domain Services dialog box.