Checking Active Directory Domain Controller Health and Replication

https://woshub.com/check-active-directory-health-and-replication/

How to Check AD Domain Controller Health Using Dcdiag?

Dcdiag is a basic built-in tool to check Active Directory domain controller health. It must always be run on an **Admin Command Prompt** To quickly check the state of an AD domain controller, use the command below:

```
dcdiag /s:DC01
```

The command runs different tests against the specified domain controller and returns a state for each test (**Passed**/**Failed**).

Typical tests:

- **Connectivity** – checks if the DC is registered in DNS, establishes test LDAP and RPC connections;
- **Advertising** – checks roles and services published on the DC;
- **FRSEvent** – checks if there are any errors of file replication service (SYSVOL replication errors);
- **FSMOCheck** – checks if the DC can connect to KDC, PDC, and Global Catalog server;
- **MachineAccount** — checks if the DC account is registered in AD correctly and if the domain trust relationship is correct;
- **NetLogons** – checks the logon privileges to allow replication to proceed;
- **Replications** – checks the state of replication between domain controllers and if there are any errors;
- **KnowsOfRoleHolders** – checks the availability of the domain controllers with FSMO roles ;

- **Services** – checks if services on the domain controllers are running;
- **Systemlog** – checks if there are any errors in the DC logs;
- Etc.

Testing AD domain controllers health using dcdiag.exe

Image not found or type unknown

You can find a full description of all available dcdiag tests here.

Besides default tests, you can run additional domain controller checks:

- **Topology** – checks if KCC has generated full topology for all DCs
- **CheckSecurityError**
- **CutoffServers** – finds a DC that is not replicated since its partner is unavailable
- **DNS** – 6 DNS checks are available (`/DnsBasic`, `/DnsForwarders`, `/DnsDelegation`, `/DnsDymanicUpdate`, `/DnsRecordRegistration`, `/DnsResolveExtName`)
- **OutboundSecureChannels**
- **VerifyReplicas** – checks if the application partitions are replicated correctly
- **VerifyEnterpriseReferences**

For example, to check if DNS is working correctly on all domain controllers, use the following command:

```
dcdiag.exe /s:DC01 /test:dns /e /v
```

dcdiag dns tests

Image not found or type unknown

It will result in a summary table showing test results on how DNS resolves names on all DCs (if it is OK, you will see **Pass** in every cell). If you see **Fail**, you need to run this test against the specified DC:

```
dcdiag.exe /s:DC01 /test:dns /DnsForwarders /v
```

To get more information from domain controller test results and save it to a text file, use this command:

```
dcdiag /s:DC01 /v >> c:\ps\dc01_dcdiag_test.log
```

dcdiag log file

Image not found or type unknown

The following PowerShell command displays only a summary information on the performed dcdiag tests:

```
Dcdiag /s:DC01 | select-string -pattern '\. (.*) \b(passed|failed)\b test (.*)'
```

Dcdiag summary report powershell script

To get the state of all domain controllers, use:

```
dcdiag.exe /s:woshub.com /a
```

If you want to display only the errors you have found, use the **/q** option:

```
dcdiag.exe /s:dc01 /q
```


dcdiag failed test

In my example, the tool has detected some replication errors:

```
There are warning or error events within the last 24 hours after the SYSVOL has been shared. Failing SYSVOL repli
..... DC01 failed test DFSREvent
```

To make dcdiag automatically fix the Service Principal Names errors for the DC account, use the **/fix** option:

```
dcdiag.exe /s:dc01 /fix
```

Checking Active Directory Replication Errors Between DCs

The built-in **repadmin** tool is used to check replication in the Active Directory domain.

Here is the basic command to check AD replication:

```
repadmin /replsum
```


repadmin /replsummary - checking active directory replication

The tool has returned the current replication status between all DCs. Ideally, the **largest delta** value should be less than 1 hour (depends on the AD topology and intersite replication frequency settings), and the number of errors = 0. In my example, you can see that one of the latest replication took 14 days, but now it is OK.

To check replication for all DCs in the domain:

```
repadmin /replsum *
```

To test intersite replication:

```
repadmin /showism
```

To view the replication topology and errors (if any), run this command:

```
repadmin /showrepl
```

The command will check the DCs and return the time and date of the last successful replication for each directory partition (`last attempt xxxx was successful`).

Repadmin /showrepl - replication status. show when the domain controller last attempted to perform

To display additional replication info, use this command:

```
repadmin /showrepl *
```

To run password replication from a writable domain controller to a read-only domain controller (RODC), the **/rodcpwdrepl** option is used.

The **/replicate** option starts the replication of the specified directory partition to a specific DC immediately.

To synchronize a specified DC with all its replication partners, use the command below:

```
repadmin /syncall <nameDC>
```

To view the replication queue:

```
repadmin /queue
```

Ideally, the replication queue should be empty.

Monitoring AD replication queues (repadmin /queue)

Check when the latest backup of the current domain controller was created:

```
Repadmin /showbackup *
```

You can also check the replication state using PowerShell. For example, the following command will display all replication errors it finds in the Out-GridView table:

```
Get-ADReplicationPartnerMetadata -Target * -Partition * | Select-Object Server,Partition,Partner,ConsecutiveReplicationFailures,LastReplicationSuccess,LastRepicationResult | Out-GridView
```

Get-ADReplicationPartnerMetadata shows an replication partner metadata object for each of its repl

Image not found or type unknown

I have uploaded a PowerShell script I often use to check the replication state in AD to my GitHub repository. The script generates an HTML file and can send it by email using the Send-MailMessage cmdlet.

https://github.com/maxbakhub/winposh/blob/main/ADHealthCheck.ps1

	## Active Directory Replication Health Check Script (PowerShell)
	## Script uses repadmin to generate HTML report and sends it to admin e-mail
	#Variables
	$report_path = "C:\Report"
	$date = Get-Date -Format "yyyy-MM-dd"
	$array = @()
	#Powershell Function to delete files older than a certain age
	$intFileAge = 8 #age of files in days
	$strFilePath = $report_path #path to clean up
	#create filter to exclude folders and files newer than specified age
	Filter Select-FileAge {
	param($days)
	If ($_.PSisContainer) {}

	# Exclude folders from result set
	ElseIf ($_.LastWriteTime -lt (Get-Date).AddDays($days * -1))
	{$_}
	}
	#get-Childitem -recurse $strFilePath | Select-FileAge $intFileAge 'CreationTime' |Remove-Item
	Function send_mail([string]$message,[string]$subject) {
	$emailFrom = "sender@woshub.com"
	$emailTo = "to@woshub.com"
	$emailCC = "cc@woshub.com"
	$smtpServer = "smtp.woshub.com"
	Send-MailMessage -SmtpServer $smtpServer -To $emailTo -Cc $emailCC -From $emailFrom -Subject $subject -Body $message -BodyAsHtml
	}
	###Test all forest
	#$myForest = [System.DirectoryServices.ActiveDirectory.Forest]::GetCurrentForest()
	#$dclist = $myforest.Sites | % { $_.Servers }
	###
	###Test specific AD domain
	$Domain = "woshub.com"
	$dclist = (Get-ADDomain $Domain -Server $Domain).ReplicaDirectoryServers
	###
	$html_head = "<style type='text/css'>
	table {font-family:verdana,arial,sans-serif;font-size:12px;color:#333333;border-width: 1px;border-color: #729ea5;border-collapse: collapse;}
	th {font-family:verdana,arial,sans-serif;font-size:12px;background-color:#acc8cc;border-width: 1px;padding: 8px;border-style: solid;border-color: #729ea5;text-align:left;}
	tr {font-family:verdana,arial,sans-serif;background-color:#d4e3e5;}

	td {font-family:verdana,arial,sans-serif;font-size:12px;border-width: 1px;padding: 8px;border-style: solid;border-color: #729ea5;}
	</style>"
	foreach ($dcname in $dclist){
	###Test all forest
	#$source_dc_fqdn = ($dcname.name).tolower()
	###
	###Test specific domain
	$source_dc_fqdn = ($dcname).tolower()
	###
	$ad_partition_list = repadmin /showrepl $source_dc_fqdn | select-string "dc="
	foreach ($ad_partition in $ad_partition_list) {
	[Array]$NewArray=$NULL
	$result = repadmin /showrepl $source_dc_fqdn $ad_partition
	$result = $result | where { ([string]::IsNullOrEmpty(($result[$_]))) }
	$index_array_dst = 0..($result.Count - 1) | Where { $result [$_] -like "*via RPC" }
	foreach ($index in $index_array_dst){
	$dst_dc = ($result[$index]).trim()
	$next_index = [array]::IndexOf($index_array_dst,$index) + 1
	$next_index_msg = $index_array_dst[$next_index]
	$msg = ""
	if ($index -lt $index_array_dst[-1]){
	$last_index = $index_array_dst[$next_index]
	}
	else {
	$last_index = $result.Count
	}
	for ($i=$index+1;$i -lt $last_index; $i++){

	if (($msg -eq "") -and ($result[$i])) {
	$msg += ($result[$i]).trim()
	}
	else {
	$msg += " / " + ($result[$i]).trim()
	}
	}
	$Properties = @{source_dc=$source_dc_fqdn;NC= $ad_partition;destination_dc=$dst_dc;repl_status=$msg}
	$Newobject = New-Object PSObject -Property $Properties
	$array +=$newobject
	}
	}
	}
	$status_repl_ko = " <font face='Calibri' color='black'>Active Directory Replication Problem : "
	$status_repl_ok = " <font face='Calibri' color='black'>Active Directory Replication OK : "
	$subject = "Active Directory Replication status : "+$date
	$message = " <font face='Calibri' color='black'>The full Active Directory Replication report is available here "
	$message += $status_repl_ko
	if ($array \| where {$_.repl_status -notlike "*successful*"}){
	$message += $array \| where {$_.repl_status -notlike " *successful*"} | select source_dc,nc,destination_dc, repl_status |ConvertTo-Html -Head $html_head -Property source_dc,nc,destination_dc,repl_status
	send_mail $message $subject
	}
	else {
	$message += "<table style='color:gray;font- family:verdana,arial,sans-serif;font-size:11px;'>No problem detected</table>"

	}
	$message += $status_repl_ok
	$message += $array \| where {$_.repl_status -like "*successful*"} | select source_dc,nc,destination_dc, repl_status |ConvertTo-Html -Head $html_head -Property source_dc,nc,destination_dc,repl_status
	$message \| Out-File "$report_path\ad_repl_status_$date .html"

view rawADHealthCheck.ps1 hosted with ❤ by GitHub

powershell script: get replication health summary report

Image not found or type unknown

You can also check the state of ADDS basic services on a domain controller using the Get-Service cmdlet:

- Active Directory Domain Services (`ntds`)
- Active Directory Web Services (`adws`) – all cmdlets from the AD PowerShell module connect to this service
- DNS (`dnscache` and `dns`)
- Kerberos Key Distribution Center (`kdc`)
- Windows Time Service (`w32time`)
- NetLogon (`netlogon`)

```
Get-Service -name ntds,adws,dns,dnscache,kdc,w32time,netlogon -ComputerName dc01
```

get adds services states on a domain controller

Image not found or type unknown

So, in this article, we have shown basic tools, commands, and PowerShell scripts you can use to diagnose the health of your Active Directory domain. You can use them in all supported Windows Server versions, including the domain controllers running in the Server Core mode.

Revision #3
Created 5 January 2024 05:18:29 by ColtM
Updated 13 June 2024 02:11:00 by ColtM